Towards Multi-Modal Face Recognition in the Wild

Chao Xiong

(Ph.D, IMPERIAL COLLEGE LONDON)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

Supervisors:

Lecturer Tae-Kyun Kim, Main Supervisor

Associate Professor Shuicheng Yan, Co-Supervisor

March 2017

Declaration

I hereby declare that the thesis is my original work and it has
been written by me in its entirety. I have duly
acknowledged all the sources of information which
have been used in the thesis.
This thesis has also not been submitted for any degree
in any university previously.

Chao Xiong

March 2017

Contents

Summary

Face recognition aims at utilizing the facial appearance for the identification or verification of human individuals, and has been one of the fundamental research areas in computer vision. Over the past a few decades, face recognition has drawn significant attention due to its potential use in biometric authentication, surveillance, security, robotics and so on. Many existing face recognition methods are evaluated with faces collected in labs, and does not generalize well in reality. Compared with faces captured in labs, faces in the wild are inherently multi-modal distributed. The multi-modality issue leads to significant intra-class variations, and usually requires a large amount of labeled samples to cover the wide range of modalities. These difficulties make unconstrained face recognition even more challenging, and pose a considerable gap between laboratorial research and industrial practice. To bridge the gap, we set focus on multi-modal face recognition in the unconstrained environment in this thesis.

This thesis introduces several approaches to address the aforementioned specific challenges. Accordingly, the approaches included can be generally categorized into two research directions. The first direction explores a series of deep learning based methods in handling the large intra-class variations in multi-modal face recognition. The combination of modalities in the wild is unpredictable, and thus is difficult to explicitly define in advance. It is desirable to design a framework adaptive to the modality-driven variations in the specific scenarios. To this end, Deep Neural Network (DNN) is adopted as the basis, as DNN learns the feature representation and the classifier with reference to the specific target objective directly. To begin with, we aims to learn a part-based facial representation with deep neural networks to address face verification in the wild. In particular, the proposed framework consists of two deliberate components: a Deep

Mixture Model (DMM) to find accurate patch correspondence and a Convolutional Fusion Network (CFN) to learn the fusion of multiple patch-specific facial features. This framework is specifically designed to handle local distortions caused by modalities such as pose and illumination. The next work introduces the conditional partition of the sample space into deep learning to tackle face recognition with regard to modalities in a general sense. Without any prior knowledge of modality, the proposed network learns the hidden modalities of faces, based on which the initial sample space is partitioned so that modality-specific feature representation can be learnt accordingly. The other direction is Semi-Supervised Learning with videos to tackle the deficiency of labeled training samples. In particular, a novel Semi-Supervised Learning strategy is proposed for the problem of celebrity identification by harvesting the "confident" unlabeled samples from the vast video sources. The video context information is adopted to iteratively enrich the diversity of the initial labeled set so that the performance of learnt classifier can be gradually improved. In this thesis, all these works are evaluated with extensive experiments in the corresponding sections. The connection and difference among the three approaches are further discussed in the conclusion section.

List of Tables

List of Figures

List of Algorithms

Glossary

W The weight of a linear projection operation. 23, 26, 27, 42–45, 47, 48, 50, 51, 72, 75, 78

X The intermediate representation of sample among the layers of DNN. 26, 27, 47, 48, 50, 72, 74–76, 78

p Local patches of faces. 46, 49, 50

x The feature representation of sample. 19, 23, 27, 29, 30, 43–47, 49, 50, 74, 75, 79, 99–101, 103–108, 116

$\mathcal{J}$ The objective function or the loss to be optimized. 23, 29, 30, 45, 47, 48, 75, 78–80, 103–107

$\mathcal{L}$ Labeled set of samples. 27, 99, 101, 102, 105–107

$\mathcal{T}$ Face Track. 99, 101, 105

$\mathcal{U}$ Unlabeled set of sample. 27, 99, 101

b The bias term. 23, 26, 27, 42–45, 48, 50, 72, 75, 78, 105, 106

y The label of input sample. 19, 27, 29, 30, 44, 45, 49, 99, 103–107

Acronyms

Ae Auto-encoder. 22, 24, 46, 51

AL Adaptive Learning. vii, 30, 101, 115, 118–122, 124, 131

APEM Adaptive Probabilistic Elastic Matching. 46, 59, 60

ART Adaptive Resonance Theory. 93, 94, 97, 98

BoW Bag of Words feature representation. 38

c-CNN conditional Convolutional Neural Network. vi, 9–11, 65, 67, 72, 73, 77, 79–81, 83, 85–87, 89, 90, 128–132

CCCP Concave-Convex Procedure. 29, 120

CFN Convolutional Fusion Network. ii, 9–11, 36, 37, 41, 42, 45, 51, 53, 58, 62, 65, 130, 131

CNB Convolutional Neural Branch. 73, 75–77, 79, 83

CNN Convolutional Neural Network. iv–vi, 3, 7, 9–11, 22, 25–27, 36, 37, 41–43, 55, 67, 68, 72, 73, 75, 77, 78, 80, 83, 86, 130, 131

DMM Deep Mixture Model. i, ii, iv, v, 10, 35–37, 41, 44, 46–49, 51, 53, 55, 58, 62, 63, 65, 127, 129–131

DNN Deep Neural Network. i, 7, 9, 10, 15, 21, 22, 69, 79, 129, 132

FV Fisher Vector. 39, 59

GPU Graphics Processing Unit. 62

HOG Histogram of Oriented Gradients. 66

LapSVM Laplacian Support Vector Machine. xii, 20, 29, 30, 103–105, 108, 109, 115, 116, 118–121, 124, 129

LBP Local Binary Pattern. iii, 4, 6, 17, 18, 34, 38, 39, 56, 59, 66, 83, 114

LFW Labeled Faces in the Wild dataset. v, vi, xi, 7, 8, 33, 37, 39–41, 45, 52, 53, 57, 58, 62, 80, 85–88

MLP Multi-Layer Perceptron. 25, 70

MPT Modality-aware Projection Tree. vi, 73–75, 77

PCA Principal Component Analysis. 16, 17, 71, 83

RKHS Reproducing Kernel Hilbert Space. 97, 103, 104

ROC Receiver Operating Characteristic. v, 56, 57, 60, 61

SAe Stacked Auto-encoder. iii, 22–26

SGD Stochastic Gradient Descent. 22, 80

SIFT Scale-Invariant Feature Transform. 4, 6, 17, 18, 34, 38, 39, 59, 66, 83, 114

SSL Semi-Supervised Learning. ii, 8, 9, 15, 20, 27, 28, 93, 95, 96, 101, 119, 128, 129, 132

SVM Support Vector Machine. xi, 4, 20, 29, 50, 106–108, 116–118

TSVM Transductive Support Vector Machine. xii, 20, 29, 30, 119–121, 129

YTF Youtube Faces dataset. v, 8, 33, 37, 52, 53, 55–58, 62

Chapter 1

Introduction

With the wide adoption and development of digital photographic and recording devices in recent years, face recognition has been one of the most promising biometric options to identify human individuals. Compared with traditional physical or virtual tools for authentication such as token and PIN, biometric traits are less likely to be misplaced, forgotten, stolen or forged. Many biometric traits have high requirements on the equipments, e.g., specific high-precision sensors for iris and fingerprint, multiple distributed cameras for body texture in person identification. Face recognition, in contrast, sets target on face captured with low-cost single camera. Face recognition addresses the problem of associating the appearance of faces with the corresponding identities. In general, the application scenarios of face recognition can be categorized into face identification and face verification. Face identification aims at predicting the identity of a given face image. Face verification, on the other hand, takes a pair of face images[1] as input and determines whether they share the same identity. Due to the increasing demand for

[1] A more general definition of face verification also takes into account the matching problem of single face to set, set to single as well as set to set.

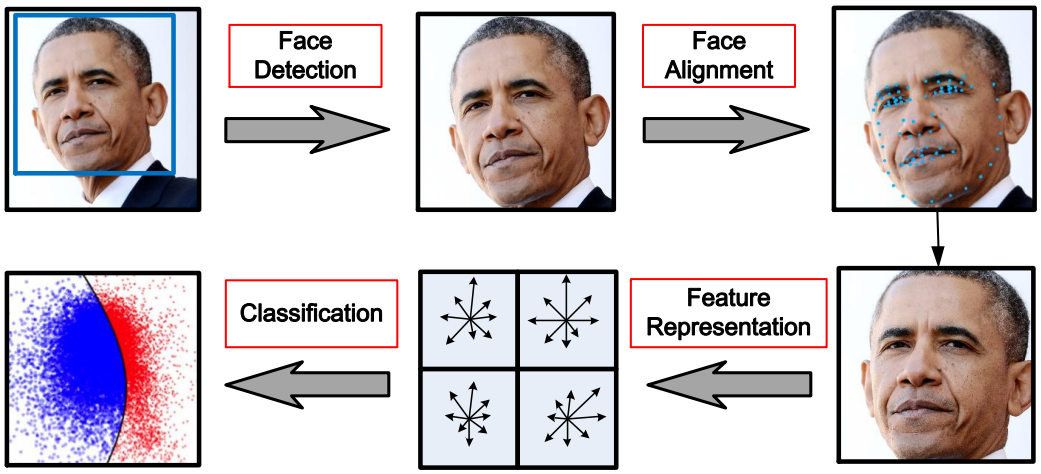

Figure 1.1: Typical pipeline of face recognition. The block diagram includes the four major components – face detection, face alignment, feature extraction and classification. The corresponding arrows indicate the common sequential order of execution.

security and great potential of application, face recognition has been an active field of research for decades. The development of related methods stimulates wide adoption in many areas including biometric authentication, surveillance, robotics, health care, human-computer interaction, multimedia analysis, etc.

Face recognition system can be typically decomposed into four key components – face detection, face alignment, feature representation and classification as shown in Figure 1.1. Many researchers do consider the integration of several components in previous works. For examples, Chen et al. [Chen et al., 2014] and Zhu and Ramanan [Zhu and Ramanan, 2012] address face detection and face alignment jointly; deep learning based methods [Sun et al., 2013b, Sun et al., 2014b, Taigman et al., 2014] integrates feature and classifier in a unified framework. However, these works still remain the concept of the aforementioned components in the frameworks. Therefore, the basics of these components are described separately as follows without loss of generality.

- **Face Detection**. Given an input image, a face detector is responsible for the localization of faces. The problem of detection is usually formulated as binary classification of facial and non-facial regions. The final output is the key facial region cropped from the original image such that the interference of the complex background texture is alleviated to the minimum. Common methods for face detection include boosting [Viola and Jones, 2001], CNN [Farfade et al., 2015] and Deformable Part Model [Yan et al., 2014]. Many face detectors are designed for only frontal or near-frontal faces of good quality, thus are sensitive to pose. Accordingly, several approaches are proposed for view-based detectors [Huang et al., 2007a, Li et al., 2002] to address this issue.

- **Face Alignment**. This step applies a landmark detector on the cropped face to determine the locations of key landmarks, such as eyes, noses and mouth. Based on the landmarks, the face image is transformed to the canonical view to suppress the impact of variations in terms of pose to a certain degree. Cascaded regression based methods [Zhou et al., 2013, Yan et al., 2013] are widely adopted for facial landmark detection, and achieve the state-of-the-art performance in the 300-W challenge [Sagonas et al., 2013]. Similar to face detectors, the variations in the unconstrained environment pose great challenges for most landmark detectors. Variation specific approaches have been proposed to deal with certain variations directly – Yang et al. [Yang et al., 2015] utilized head pose to regularize the detection of landmarks; Ghiasi and Fowlkes [Ghiasi and Fowlkes, 2014] proposed a face landmark detector that models occlusions of parts explicitly.

- **Feature Extraction**. The face images represented in raw-pixel format are usually not discriminative enough for later classification step. Therefore, further processing is usually required to extract the salient information which is sensitive to variations across different identities, while robust to geometric and photometric variations at the same time. Ideal features should be discriminative to distinguish

different persons with low dimensions. Local Binary Pattern (LBP) [Ojala et al., 1996] and Scale-Invariant Feature Transform (SIFT) [Lowe and G, 1999] are two examples of manual features typically used for feature extraction in face recognition.

- **Classification**. In the step of classification, a cognitive model, termed as classifier, is learnt to make decisions based on the similarity between a testing image and the given set of training images. The output of such a classifier is the identity of the testing image for face identification and a "yes" or "no" answer for face verification. Many typical classifiers in machine learning have been successfully applied in the problem of face recognition, including Support Vector Machine (SVM) [Li et al., 2013], random forest [Kouzani et al., 2007], bayesian classifier [Chen et al., 2012], etc.

When referring to face recognition in general, the design of feature representation and recognition classifier are usually the primary focus of research. Correspondingly, this thesis tackles the problem of face recognition from these two perspectives.

Early research attempts [Turk and Pentland, 1991, Belhumeur et al., 1997] on face recognition are mainly examined on faces under the controlled laboratory settings. Under such settings, only a few chosen modalities are taken into consideration when constructing the databases. Accordingly, the proposed methods are only robust to the predefined modalities in the given datasets, thus may not generalize well in the real-world environment. The term "modalities" usually refers to different sensory input channels, such as photo, infra-red, sketch, text, sound, etc. Different modalities depict an object from different independent perspectives. Similar to [Sharma and Jacobs, 2011, Mignon and Jurie, 2012, Kim and Kittler, 2005], this thesis extends the meaning of modality further to the independent photographic factors such as pose, illumination, resolution, etc. In general, we define the term "modalities" as all possible factors that may cause con-

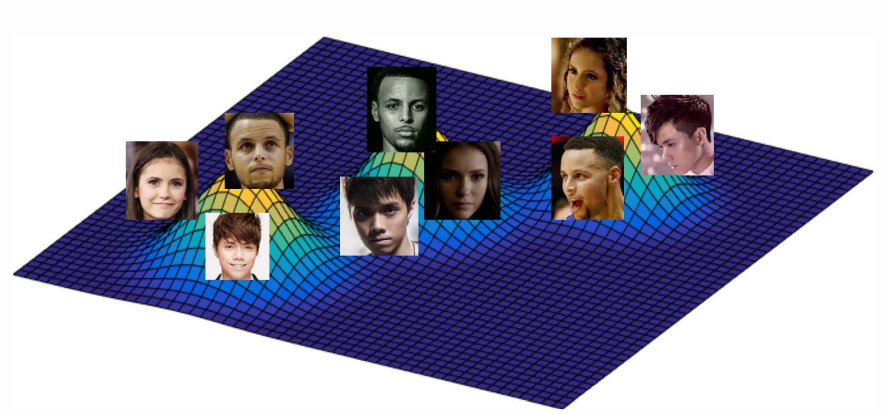

Figure 1.2: Spare distribution of same-ID faces in terms of modalities.

siderable within-class differences such that faces of the same identities are sparsely distributed in the resulting feature space. In other words, faces of the same modality may be lying close to each other regardless of the identity as shown in Figure 1.2. With the growing demand for real-world applications, face recognition in the wild becomes more and more crucial, and draws much attention from both academic and industrial aspects. In the uncontrolled environment, the photographic conditions are complex and unpredictable, and it is common to observe the co-existence of various modalities. In most cases, the recognition of faces in the wild is a multi-modal classification problem, which raises much more challenges and requirements for face recognition systems. Thanks to the persistent research efforts, the area of multi-modal face recognition has made great progress in the past a few decades. However, the problem still remains quite challenging for most existing methods due to multiple difficulties.

In this thesis, we propose several approaches to further address these difficulties for unconstrained face recognition with multi-modalities. The rest of this chapter is organized as follows. Section 1.1 briefly introduces the major challenges caused by multi-modalities. Section 1.2 afterwards gives an overview and summarizes the main contributions of this thesis in handling these challenges.

1.1 Challenges

In reality, the issue of multi-modalities is usually inevitable for face recognition in the wild. Consequently, most unconstrained face recognition problems are inherently multi-modal distributed. Typical examples of modalities include illumination, pose, facial expression, age and occlusion as shown in Figure 1.3. The essence of inherent multi-modalities can cause difficulties from the following two perspectives.

1.1.1 Significant Intra-class Variations

In the ideal case, faces of the same identity should be lying close to each other in the feature space. However, the existence of multi-modalities renders this assumption invalid in the uncontrolled environment. Face images in the wild are usually captured with a wide range of modalities, and the combination of modalities in a specific problem is complex and thus hard to predict. Multi-modality results in large intra-class variations for faces of the same identity, and raises great challenges to most existing methods.

To counteract the impact of modalities, many features are manually designed in the early works of face recognition. Typical hand-crafted features include Local Binary Pattern (LBP) [Ojala et al., 2002b], Scale-Invariant Feature Transform (SIFT) [Lowe and G, 1999], Gabor feature [Daugman, 1985], etc. These hand-crafted features are designed for the generic purposes of handling certain kinds of variations. However, since the modalities involved vary case by case, such pre-defined hand-crafted features may not generalize well to a specific problem. Moreover, the process of quantization is usually included in the extraction of hand-crafted features to reduce the resulting dimension. Some crucial information may be eliminated during quantization, and cannot be recovered in the following classification procedure. Ideally, the feature representation should be robust to the specific modality-oriented variations in the given scenario. Moreover,

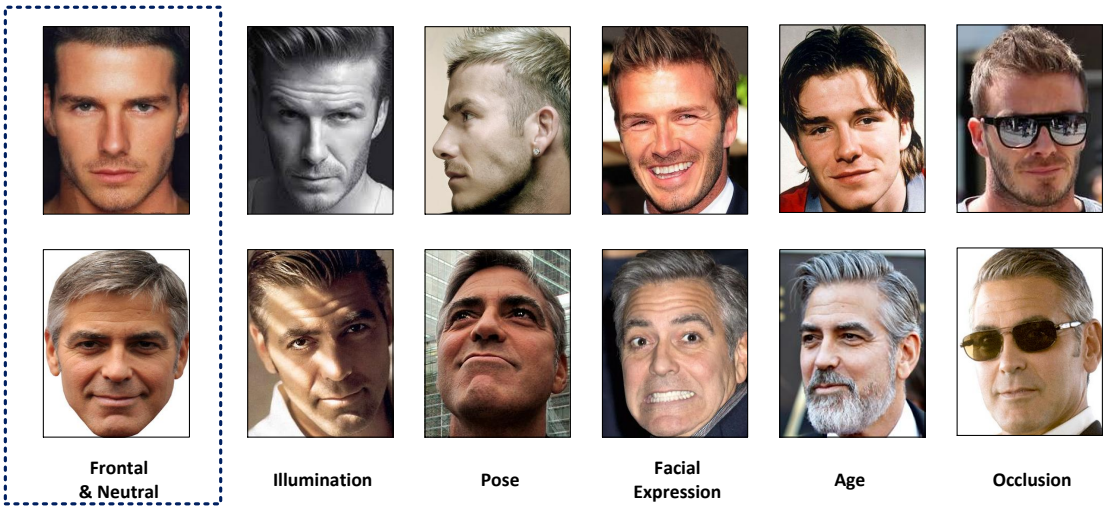

Figure 1.3: Facial variations with regard to modalities. Images in the same row share the same identity. For each ID, five kinds of variations are illustrated to demonstrate the significant appearance differences in terms of modalities. The categories of modality are listed below the corresponding column.

the designs of feature and classifier are usually studied separately. Different combinations of feature and classifier may result in different performance, which makes the problem even more complex and difficult to handle in practice.

The breakthrough of deep learning brings about a possible solution of learning the feature representation and classifier in a joint manner. The advantage of deep learning is that the feature and classifier are learnt with the direct guidance of the target objective. Thus, the learnt feature and classifier are optimal for the given problem. Recent efforts on deep learning have achieved great successes in various fields of computer vision [Krizhevsky et al., 2012, Farabet et al., 2013, Huang et al., 2012b, Nair and Hinton, 2010,Sun et al., 2013a]. Deep Neural Network (DNN), especially the Convolutional Neural Network (CNN), has also been proven effective for the problem of unconstrained face recognition. In particular, the state-of-the-art performance on the benchmark database for face verification LFW has been improved by a series of deep learning based meth-

ods [Hu et al., 2013, Taigman et al., 2014, Sun et al., 2014b] over and over again just in a few years. Accordingly, we propose a series of methods built on deep learning to deal with multi-modalities from two perspectives in Chapter 3 and Chapter 4.

1.1.2 Scarcity of Labeled Samples

Many existing approaches for face recognition are supervised, i.e., they rely on a set of labeled samples to make inference on the unseen data. The bottleneck of such supervised methods usually lies in the scale of the labeled training set. To deal with the wild range of modalities in reality, an ideal solution is to collect as many labeled samples as possible to cover all possible modalities. With such a dataset, it would be much easier to design a generic framework that generalizes well to most real-world tasks. However, supervised data or labeled data are difficult and expensive to acquire, because it usually needs significant manual efforts of the annotators. In spite of the difficulty, many researchers have attempted to constructing such large databases of labeled faces. Typical examples of such databases include Labeled Faces in the Wild (LFW) [Huang et al., 2007b], Youtube Faces (YTF) [Wolf et al., 2011], Pubfig [Kumar et al., 2009], etc. However, the number of labeled samples is still far from enough to cover all kinds of variations in reality.

Compared with labeled face samples, unlabeled samples are easier to access and usually arrive in a large volume. The existence of commercial search engines and video sharing websites lowers down the amount of manual labor for unlabeled data collection to a large extent. It is reasonable to assume that vast unlabeled samples reveal the underlying distribution in the sample space which provides crucial clues for transferring knowledge from the labeled to the unlabeled. Many works built on such an assumption is categorized as Semi-Supervised Learning (SSL). SSL attempts to take the labeled data points as seeds and utilize readily available unlabeled data points to improve the

recognition accuracy of the classifier. Due to its practical significance, SSL has been successfully applied in many previous works [Belkin et al., 2006, Zhu et al., 2003a, Cherniavsky et al., 2010]. Inspired by the idea of SSL, this thesis introduces a semi-supervised framework in addressing the deficiency of labeled samples in Chapter 5.

1.2 Contributions and Outline

In this thesis, we introduce several approaches for the problem of unconstrained face recognition, which takes into account both aforementioned challenges caused by multi-modalities. Accordingly, these approaches can be categorized into two general directions corresponding to those challenges.

To begin with, we propose two frameworks built on the basis of CNN to address the variations with regard to multi-modalities. The two approaches study the cross-modality face recognition problem from two different perspectives. The first approach gets the inspiration from the success of patch-based methods with hand-crafted features. In this work, we propose a Convolutional Fusion Network (CFN) integrating the merits of DNN into the construction of part-based facial representation. The proposed framework sets focus mainly on the local distortions caused by modalities such as pose and illumination. The second work, on the other hand, presents a conditional Convolutional Neural Network (c-CNN) in addressing the multi-modal problem in a more general sense. In contrast with conventional approaches for cross-modality problems, the modalities are unknown but learnt as crucial clues to partition the data for better generalization performance.

Secondly, a Semi-Supervised Learning based approach is introduced in tackling the issue of insufficient labeled samples for celebrity identification in videos. In particular, the video context is incorporated into the self-training theme such that the proposed

method gradually improves the diversity of the training set. As a result, the learnt classifier is capable of evolving with an initial labeled set with only a limited number of samples.

It is also worthy mentioning that the two research directions are not mutually exclusive. The semi-supervised learning theme can be easily extended via utilizing DNN as a classifier trained directly on the raw-pixel inputs. Furthermore, the two deep learning methods are correlated as well. To be more specific, c-CNN can be adopted to replace CNN in CFN to generate a part-based representation. The rest of the thesis is organized as follows.

Background (Chapter 2)

This chapter gives a brief overview of the algorithms related. We firstly introduce some typical methods used for generic face recognition. Corresponding to the two directions of research, the basics of semi-supervised learning and deep learning are also included.

Part-based Deep Facial Representation (chapter 3)

In this chapter, we propose to learn a part-based feature representation under the supervision of face identities through a deep model, which ensures the generated representations are more robust and suitable for face verification. The proposed framework consists of the following two deliberate components: a Deep Mixture Model (DMM) to find accurate patch correspondence and a Convolutional Fusion Network (CFN) to extract the part based facial features. Specifically, DMM robustly depicts the spatial-appearance distribution of patch features over the faces via several Gaussian mixtures, which provide more accurate patch correspondence even in the presence of local distortions. Then, DMM feeds only the patches which preserve the identity information to the

following CFN. The proposed CFN is a two-layer cascade of Convolutional Neural Networks (CNN): 1) a local layer built on face patches to deal with local variations and 2) a fusion layer integrating the responses from the local layer. CFN jointly learns and fuses multiple local responses to optimize the verification performance. The composite representation obtained possesses certain robustness to pose and illumination variations and shows comparable performance with the state of the arts on two benchmark data sets.

- **C. Xiong**, L. Liu, X. Zhao, S. Yan and T-K. Kim, *Convolutional Fusion Network for Face Verification in the Wild*, Accepted to appear in IEEE Trans. on Circuits and Systems for Video Technology, 2015

Generic Cross-Modality Face Recognition (Chapter 4)

This chapter proposes a conditional Convolutional Neural Network, named as c-CNN, to handle the generic problem of multi-modal face recognition. Different from traditional CNN that adopts fixed convolution kernels, samples in c-CNN are processed with sets of kernels dynamically activated. In particular, convolution kernels within each layer are only sparsely activated when a sample is passed through the network. For a given sample, the activations of convolution kernels in a certain layer are conditioned on its present intermediate representation and the activation status in the lower layers. The activated kernels across layers define the sample-specific routes that reveal the distribution of underlying modalities. Consequently, the proposed framework does not rely on any prior knowledge of modalities in contrast with most existing methods. To substantiate the generic framework, we introduce a special case of c-CNN via incorporating the conditional routing of the decision tree, which is evaluated with two problems of multi-modality – multi-view face identification and occluded face verification. Exten-

sive experiments demonstrate consistent improvements over the counterparts unaware of modalities.

- **C. Xiong**, X. Zhao, D. Tang, K. Jayashree, S. Yan and T-K. Kim. *Conditional Convolutional Neural Network for Modal-aware Face Analysis*. Proc. of IEEE Int. Conf. on Computer Vision (ICCV), Santiago, Chile, 2015

Semi-Supervised Learning with Video Context (Chapter 5)

In this chapter, a novel semi-supervised learning strategy is proposed to address the problem of celebrity identification. The video context information is explored to facilitate the learning process based on the assumption that faces in the same video track share the same identity. Once a frame within a track is recognized confidently, the label can be propagated through the whole track, referred to as the confident track. More specifically, given a few static images and vast face videos, an initial weak classifier is trained and gradually evolves by iteratively promoting the confident tracks into the "labeled" set. The iterative selection process enriches the diversity of the "labeled" set such that the performance of the classifier is gradually improved. This learning theme may suffer from semantic drifting caused by errors in selecting the confident tracks. To address this issue, we propose to treat the selected frames as related samples – an intermediate state between labeled and unlabeled instead of labeled as in the traditional approach. To evaluate the performance, a new dataset is constructed with 3000 static images and 2700 face tracks of 30 celebrities. Comprehensive evaluations on this dataset and a public video dataset indicate significant improvement over established baseline methods.

- **C. Xiong**, G. Gao, S. Yan, Z. Zha, H. Ma and T-K. Kim, *Adaptive Learning for Celebrity Identification with Video Context*, IEEE Trans. on Multimedia, Vol.16, No.5, Aug 2014

Conclusions and Future Works (Chapter 6)

This chapter summarizes the conclusions drawn from the previous chapters and further discusses the correlation across different methods included. Finally, a discussion on potential directions of research is given for future works.

Chapter 2

Background

In this chapter, a brief literature review is given on research areas related to the methods in this thesis. To begin with, basics of the standard face recognition system are introduced. However, such standard systems are not capable of handling the difficulties in the problem of multi-modal face recognition. To solve the issues, we propose several algorithms falling into two broad directions of research – deep learning and Semi-Supervised Learning. Accordingly, Section 2.2 reviews some crucial key-points for Deep Neural Network, and Section 2.3 introduces some of the standard algorithms for Semi-Supervised Learning. This chapter introduces the corresponding research fields only to give a brief understanding of the target problem, and the comparisons with specific works are further explained in the corresponding main chapters with more details.

2.1 Generic Face Recognition

As mentioned in Chapter 1, a standard face recognition system includes four fundamental components, i.e., face detection, face alignment, feature extraction and classification. For the topic of face recognition, feature representation and recognition classifier play a crucial role, and thus draw much attention.

2.1.1 Feature Extraction

In many applications, the raw pixel representation is adopted for faces images, i.e., face images are represented as a 3-D matrix composed of the intensity values of pixels. The raw pixel representation contains much redundancy and is sensitive to variations in terms of modalities such as pose, lighting, occlusion, etc. Therefore, the raw face images are usually pre-processed by various feature extraction methods. An ideal feature representation should be robust to both holistic and local variations while retaining the identity-preserving information with minimal physical memory. In general, existing features can be categorized as holistic representation and local representation.

Holistic representation aims at projecting the full face images into a target subspace so as to eliminate noisy components irrelevant to identity. Eigenface [Turk and Pentland, 1991] is one of the earlist approaches for holistic feature representation. The major component of Eigenface is Principal Component Analysis (PCA) [Person, 1901], which learns the eigen vectors, termed as eigenfaces, of the covariance matrix computed from the training face set. Each face image is reconstructed as a linear combination of eigenfaces. The reconstruction coefficients are then used as the representation of the corresponding faces. The similar idea of subspace representation is also adopted in the work of Fisherfaces [Belhumeur et al., 1997]. More recently, sparse representation, also known as sparse coding, is also utilized for the problem of face recognition [Yang et al.,

2007, Yang et al., 2011]. In these methods, the testing face is represented as a linear combination of dictionary entries. Different from PCA, a sparse constraint is posed on the combination coefficients via ℓ_1 norm regularization. However, holistic representation is usually expensive in computation of features, thus is not scalable for large-scale problems.

Local feature extracts the characteristics from partial face regions, and demonstrates great robustness to local variations. Two examples of local representation are listed as follows.

- **Local Binary Pattern (LBP)**. LBP [Ojala et al., 2002b] is a representative approach for local feature extraction. Due to its effectiveness and low computation cost, LBP has been widely used in many face recognition problems [Chen et al., 2013,Li et al., 2013,Zhu et al., 2015]. The purpose of LBP is to encode the local contrast information into a histogram of texture patterns. In particular, LBP compares the central pixel of every local patch with its adjacent or surrounding pixels, and assigns a binary label for each comparison. The binary sequence for each pixel is then transformed to a decimal digit which is then used as a bin of the final histogram. The corresponding procedure of LBP extraction is illustrated in Figure 2.1.

  Due to the success of LBP, many variants have been proposed for further improvement. Typical examples include Dynamic Texture [Zhao and Pietikainen, 2007], Multi-scale Block LBP [Liao et al., 2007], Locally Assembled Binary Haar feature [Yan et al., 2008], Local gabor binary pattern [Zhang et al., 2005] and so on.

- **Shift-Invariant Feature Transform (SIFT)**. SIFT [Lowe and G, 1999] is also a widely used local features for face recognition [Li et al., 2013, Simonyan et al., 2013, Chen et al., 2013]. The advantage of SIFT lies in its invariance to scaling, rotation and translation of images. SIFT also considers the contrast between a pixel and its surrounding ones as the key factor in representing a face image. Differ-

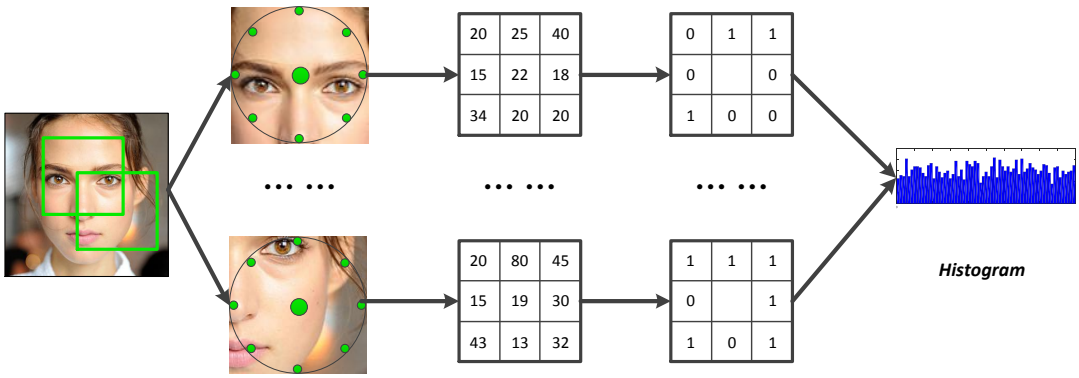

Figure 2.1: Illustration on the feature extraction procedure of LBP. The face image on the left is firstly divided into multiple sub-blocks. Each sub-block is scanned through pixel by pixel. The texture description, illustrated as the 3x3 block, is then computed by comparing the central pixel with its surroundings. After all the sub-blocks are processed, a histogram is calculated to be the final feature.

ent from LBP, standard SIFT firstly localizes the key points, i.e., the high-contrast corner points, by detecting the extrema across scale and space. The descriptor for each key point is then extracted by computing the orientation and magnitude of gradients in its 16x16 neighboring region. The neighboring region is further divided into 16 sub-blocks of size 4x4, in each of which a 8 bin orientation histogram is computed. Accordingly, the dimension of the final histogram is 128. There are also variants that directly apply the SIFT descriptor to represent the face images as Dense SIFT in [Hu et al., 2013, Simonyan et al., 2013].

In the application of local descriptors, it is inevitable to consider the compromise between informativeness and efficiency. Once certain information is lost in local quantization, it cannot be recovered in the following process. Besides, the design of such features usually requires enormous manual efforts and expertise in research, thus they are not adaptive or optimal for specific problems.

2.1.2 Classification for Face Recognition

Classification aims at learning a model to make predictions on the unseen data based on previous observations. Many classifiers are learnt in a supervised manner, i.e., the previous observations or samples come with ground-truth labels. The generalization performance of the supervised classifier, therefore, largely relies on the sufficiency of labeled training samples. Some of the typical supervised classifiers are introduced in the following.

- **Nearest Neighbor Classifier(NN)**. 1-N-N classifier is probably one of the simplest classifier in machine learning. In 1-N-N classification, the label of an unseen sample point is predicted the same as its nearest neighboring sample. K-N-N classifier is a popular variant of 1-N-N, which makes the prediction by majority voting based on the label distribution of the k nearest neighboring points instead. The concept of "NEAR" is usually defined with regard to the distance measurement in the feature space. Therefore, the performance of nearest neighbor classifier relies on the effectiveness of feature representation to a large extent.

- **Naive Bayesian Classifier**. Naive Bayesian classifier [Domingos and Pazzani, 1997] falls into the general category of probabilistic classifier based on the Bayes' theory. Bayesian classifier relies on a simple assumption that dimensions of the feature vector are independent of each other. Following the Bayes' theorem, the posterior probability of the classification problem can be formulated as follows.

$$P(y_k|\mathbf{x}_i) \propto P(y_k) \cdot \prod_{j=1}^{n} P(\mathbf{x}_i^{(j)}|y_k), \tag{2.1}$$

  where $P(\cdot)$ represents probability and y_k is the label of class k. Accordingly, the prediction is made by choosing the label of class which gives the highest posterior probability.

- **Classification Forest**. Classification forest is a specific case of random forest [Breiman, 2001] applied on the problem of classification. Random forest involves with the concept of ensemble learning via training a large number of decision trees, each of which acts as a weak classifier. The application of bagging leads to a certain level of independence among the trees, which brings a significant improvement after fusing the results of all the trees. During the testing process, an unseen sample is examined against a series of simple split rules of tree nodes along the path, and finally falls into a leaf node. The class distribution in the corresponding leaf node is taken to compute the posterior probability. Finally, the prediction is made via averaging the results of each tree. The label of class with the highest probability is then assigned to the unseen sample. Classification forest is highly scalable to the classification problem of large scale, and has been widely applied to various fields of research [Gall and Lempitsky, 2013, Fette et al., 2007, Bosch et al., 2007].

- **Support Vector Machine (SVM)**. SVM is firstly introduced by Boser et al. [Boser et al., 1992] as a non-probabilistic classifier for binary classification. The extension of SVM to multi-class classification is proposed later in [Vapnik and Vapnik, 1998]. The motivation of SVM is to learn a decision surface that separates the training samples via maximizing the decision margin. The kernel trick enables SVM to be generally applied to the non-linear classification problems as well. The good generalization performance makes SVM one of the most popular classifiers. The concept of maximal margin is also extended to the realm of Semi-Supervised Learning. Typical examples include Laplacian Support Vector Machine (LapSVM) and Transductive Support Vector Machine (TSVM), which are two basis algorithms used in Chapter 5.

In many previous works, the choices of features and classifiers are determined empiri-

cally or by extensive experiments. The complex combination of these two crucial components remains a difficult issue when deploying face recognition in the specific scenarios. To address this issue, researchers have been attempting to learn the feature and classifier in a joint manner. Deep learning, or Deep Neural Network, is one of the representative methods following this idea.

2.2 Deep Neural Network

The rapid development of Internet-oriented applications, such as video sharing websites, search engines and social networks, results in the explosion of data in quantities. The vast sources of data brings the human society to the era of big data. Big data brings both challenges and opportunities for the academic fields of computer vision and machine learning. Simulated by this trend, deep learning, also known as the Deep Neural Network (DNN), emerges as a new research of interest in the past few years, and draws much attention from both academia and industry. The work of Hinton and Salakhutdinov [Hinton and Salakhutdinov, 2006] has inspired a large number of researches and simulated applications in many fields [Krizhevsky et al., 2012, Farabet et al., 2013, Huang et al., 2012b, Nair and Hinton, 2010, Sun et al., 2013a].

Compared with the pipeline of standard face recognition systems, deep learning integrates the learning of feature representation and classifier in a joint manner. The learning process is directly conducted with regard to the objective of the given problem, thus the learnt feature is optimal for the target. Moreover, DNN adopts a cascaded structure of multiple feature extraction layers – the intermediate representation of the lower layer is forwarded to the upper layer as inputs. In most cases, a non-linear activation function is applied on the outputs of each layer for better generalization. In contrast to metric learning [Cui et al., 2013, Davis et al., 2007, Guillaumin et al., 2009] with shallow structure of linear transformations, the deep cascade of non-linear pro-

jections provides DNN with higher level of discriminative capability and abstraction, which has been proven effective in many recent works.

DNN introduces a generic solution to the problem of classification. It does not require problem-specific pre-processing or feature extraction of data. In most cases, researchers utilize the raw data as the input of the network. The learning of the network is achieved by alternating between the forward propagation and back propagation repeatedly until convergence. In the forward propagation, the inputs are passed through the network layer by layer, and a cost function or objective is computed based on the outputs of the final layer in current iteration. The gradients of the cost are then computed with regard to the weights and the intermediate inputs of each layer, and further propagated to the former layer according to the chain rule. For the back-propagation steps, Stochastic Gradient Descent (SGD) is a common option for updating the parameters.

Various architectures have been proposed for deep learning in recent years. In this section, we give a brief introduction of two widely-used structures which are directly related to Chapter 3 and Chapter 4 – Stacked Auto-encoder (SAe) for unsupervised learning and Convolutional Neural Network (CNN) for supervised learning.

2.2.1 Stacked Auto-encoder

Stacked Auto-encoder (SAe), in general, is an unsupervised deep neural network, and is usually used in the phase of pre-training. Pre-training is a common pre-processing step in training DNN. The purpose is to locate the parameters to a good initial position in the parameter space for later supervised fine-tuning process.

The basic building block of SAe is Auto-encoder (Ae). An Ae consists of an encoder layer and an decoder layer – the encoder layer defines a linear mapping to the feature space; the decoder layer defines a linear mapping to the original space. The input vector

is forwarded to the encoder to extract low-dimensional feature. The decoder is stacked after encoder to map the feature back to the original space with reconstruction. The network is then optimized to minimize the error between the original input and the reconstructed version. The motivation is to learn an effective low-dimensional representation so that the encoding reserves most of the crucial information that is needed to reconstruct the original sample.

In particular, the forward function of the encoder layer can be formulated as

$$\mathbf{a} = \sigma(\mathbf{W}\mathbf{x}_i + b), \tag{2.2}$$

where $\mathbf{a}$ is the representation of $\mathbf{x}_i$, $\mathbf{W}$ and b refer to the weight and bias that need to be learnt, and $\sigma(\cdot)$ is a non-linear activation function. Typical activation functions include Rectified Linear Unit [Nair and Hinton, 2010] $\sigma(x) = max(x, 0)$, logistic activation $\sigma(x) = \frac{1}{1+e^{-x}}$ and tanh activation $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

As for the decoder layer, the forward function is

$$\widetilde{\mathbf{x}}_i = \sigma(\mathbf{W}'\mathbf{x}_i + b'). \tag{2.3}$$

In many auto-encoder networks, the strategy of tied weights is adopted, i.e., $\mathbf{W}' = \mathbf{W}^T$. The objective, accordingly, is defined as the Euclidean distance between $\widetilde{\mathbf{x}}_i$ and $\mathbf{x}_i$ as follows,

$$\mathcal{J} = ||\widetilde{\mathbf{x}}_i - \mathbf{x}_i||_2 \tag{2.4}$$

SAe is constructed via stacking encoding and decoding layers of auto-encoder respectively. Specifically speaking, the input of a SAe is encoded by a cascade of the encoder layers. Accordingly, the decoder of SAe is formed by placing the corresponding decoder layers in the reverse order. An illustration is given in Figure 2.2 as an example. During the training of SAe, each encoder-decoder pair is trained separately in order (from Ae_1 to Ae_2). The input of each encoder-decoder pair is the intermediate feature $\mathbf{z}$

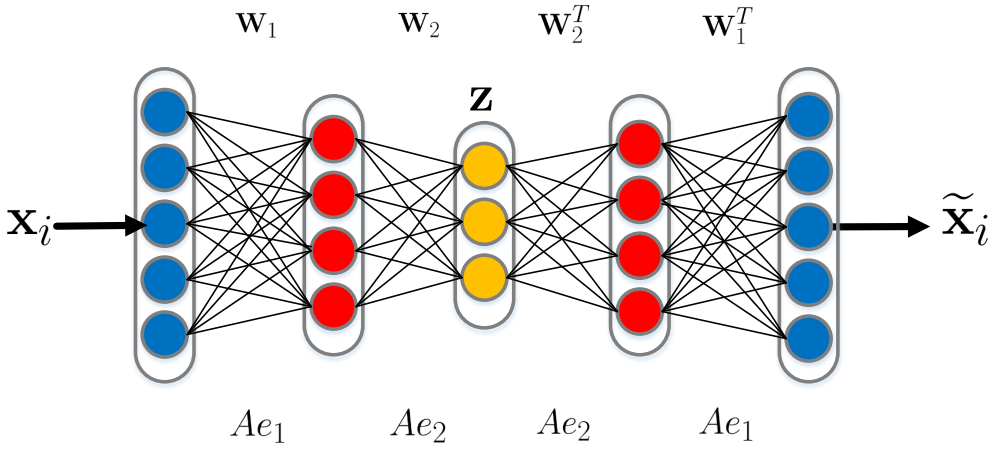

Figure 2.2: Structure of SAe. SAe is built by stacking multiple auto-encoders together. The encoder and decoder layer of each auto-encoder is placed in pairs, the corresponding neutrons are illustrated with the same color. In this graph, the weights of encoder and decoder are tiled, i.e., the weight matrix of the decoder is the transpose of that of the corresponding encoder.

produced by the previous Ae, except for the first Ae which is trained with the original raw input.

After the training of SAe, the encoder part is usually taken as a feature extractor which gives an proper initialization. Therefore, the encoder layers are inherited into the fine-tune network with a supervised classification layer attached to the end as in [Hinton and Salakhutdinov, 2006].

2.2.2 Convolutional Neural Network

Fully-connected layer is one of the most common layers used in deep neural network. The name comes from the fact that each neutron, the basic component of a layer, in one layer is fully connected with each neuron in the next layer. Namely, there is a connection with weight for every neuron pair in two adjacent layers. Such fully-connected layer is designed for generic usage.

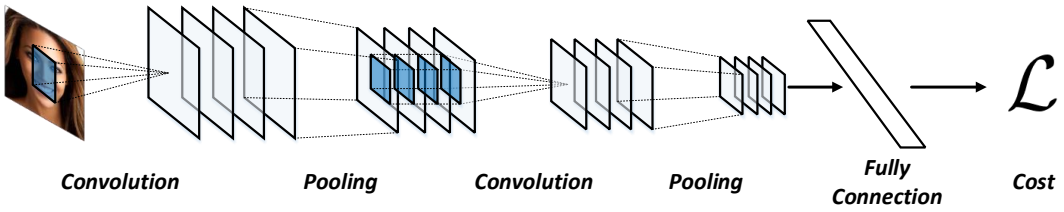

Figure 2.3: Typical structure of CNN. CNN is composed of several convolution layers and fully connected layers. In addition, it is common to include a spatial pooling layer right after the convolution layer. The whole network is learnt via back-propagation from the layer in the back, i.e., the loss layer, to the front layers.

Convolutional Neural Network (CNN) is designed primarily for the analysis of images, including object recognition, scene parsing, face detection, face recognition and so on. The basic module of CNN is the convolution layer which includes a large number of convolution kernels. Different from fully-connected network such as SAe and Multi-Layer Perceptron (MLP), a Convolutional Neural Network stacks multiple convolution layers for feature extraction. Convolution layer take the advantage of 2D images via exploring only the correlations among the locally adjacent pixels. Such approach is inspired by the mechanism of receptive fields in human eyes. Spatial pooling is a non-linear down-sampling process, which usually takes the non-overlapping regions from each feature map and output the mean or maximal value. In this way, pooling largely reduces the size of output feature maps and the corresponding convolution or linear operations in the following layers. Moreover, spatial pooling following the convolution layer introduces certain level of robustness to geometric translation, and thus is widely used in the design of CNNs. Figure 2.3 gives an example of typical CNN structure.

The convolution kernel in CNN conducts the convolution operation on the 3D image input (height$\times$width$\times$channels). The convolution kernel can be viewed as a locally-connected linear mapping as shown in Figure 2.4. The local connection brings signif-

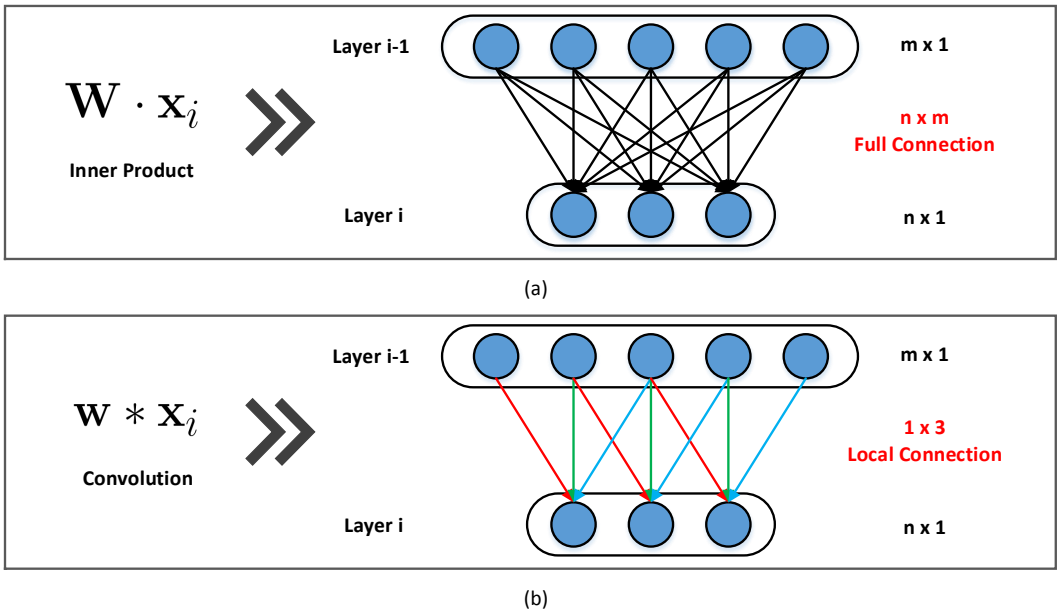

Figure 2.4: Comparison between full connection and local connection. The operation of full connection is illustrated in figure (a), and the operation of local connection is illustrated in figure (b). For full connection, each neutron pair between two adjacent layers $i-1$ and i has a connection with corresponding weight to learn. For local connection, the value of a neutron in layer i is determined only by its near-by neutrons. The kernel in this graph is of size 1×3, and the connections are illustrated as arrows with different colors. Clearly, the weight parameters 1×3 need to be learnt in local connection are much fewer than those in full connection $n \times m$.

icant advantages since it reduces the number of parameters to a large extent such that the network is easier to train and faster to converge.

In the forward process of a convolution layer, one 3-D convolution kernel is applied on all the feature maps – channels of the 3-D outputs of previous layer, and is used to compute one output feature map of the current layer. Similar to SAe, CNN also includes non-linear activations on the output of most layers. Accordingly, the forward function of convolution layer i can be formulated as

$$\mathbf{X}_{n,k}^{(i+1)} = \sigma(\mathbf{W}_k^{(i)} * \mathbf{X}_n^{(i)} + b^{(i)}), \tag{2.5}$$

where $\mathbf{X}_{n,k}^{(i)}$ is the k-th feature map of the n-th sample of layer i, and the symbol $*$ refers to the convolution operation. $\mathbf{W}_k^{(i)}$ and $b^{(i)}$ represent the kernel weight and bias of layer i respectively. The operator $*$ refers to the convolution computation. The optimization of CNN is also achieved via back-propagation based on the chain rule.

2.3 Semi-Supervised Learning

Labeled samples are difficult and expensive to acquire as the labeling process demands for enormous human efforts. In contrast with the labeled samples, unlabeled samples are easier to obtain and usually arrive in large amount. SSL [Zhu, 2005], short for Semi-Supervised Learning, aims to utilize the vast source of unlabeled samples with the guidance of labeled samples during the learning process. In general, SSL falls between supervised learning and unsupervised learning. The basic reasoning behind SSL lies in the assumption that unlabeled samples reveals the underlying distribution of the sample space. With years of research efforts, many approaches [Yarowsky, 1995, Mitchell, 1999, Zhu et al., 2003a] have been proposed to address the classification problem in a semi-supervised manner. Generic SSL methods assume the presence of a labeled set $\mathcal{L} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l\}$ and a unlabeled set $\mathcal{U} = \{\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, ..., \mathbf{x}_{l+u}\}$, where $u \gg l$. Each labeled sample $\mathbf{x}_i$ is provided with a ground-truth label y_i. In this section, we only include several representative SSL methods that are closely related to Chapter 5.

2.3.1 Self Training

Self-training [Yarowsky, 1995] can be viewed as a variant of supervised learning. In the standard self-training theme, a classifier is firstly trained on an initial labeled set, and then makes prediction on the unlabeled set. The most confidently recognized samples are then chosen from the unlabeled set, and assigned with the prediction labels. In the

following training iterations, the already chosen samples are treated the same as the labeled set to re-train or update the classifier. In a sense, self-training relies on prediction of its own classifier to gradually improve the performance. The success of self-training is largely dependent on the accuracy of chosen unlabeled samples in each iteration. However, the errors in the selection process are inevitable, and will gradually accumulate over iterations. Such errors in early iterations are fatal and will deteriorate the classifier especially when the initial labeled set is of small size. In particular, it is possible that the learnt classifier will gradually drift far away from the original meaning of the labeled instances due to the errors in selection. In literature , such phenomenon is called "Semantic Drifting". Chapter 5 proposes a Semi-Supervised Learning method similar to self-training. The problem of semantic drifting is specifically addressed via adaptive re-weighting of the selected samples.

2.3.2 Co-training

Co-training [Mitchell, 1999] also adopts the iterative selection of confident unlabeled samples during training. Different from self-training relying on only one classifier, co-training involves two classifiers trained on two-views of data. In each iteration, the two classifiers are trained with their corresponding view or feature representation of labeled data, and make predictions on the unlabeled samples. Based on the confidence scores, each classifier provides recommendations on which sample should be promoted as labeled sample for the other classifier. The main drawback of co-training lies in the harsh requirements on the data – data should be represented in two views; the two views should be conditional independent; the data of each view should be sufficient to train an accurate classifier. Similar to self-training, errors will also accumulate during the selection of confident samples.

2.3.3 Transductive Support Vector Machine

Transductive Support Vector Machine (TSVM) aims at integrating the discriminative decision boundary and the underlying data distribution in a unified cost. The general idea is to locate the decision boundary in the low-density zone of the sample space, i.e., the partition of data should avoid passing through the high-density area. TSVM makes modification to the standard SVM by imposing an extra constraint on the unlabeled samples. A common objective function of TSVM is formulated as

$$\mathcal{J} = \sum_{i=1}^{l} max((1 - y_i f(\mathbf{x}_i)), 0) + \gamma_A ||f||_A^2 + \gamma_I \sum_{i=l+1}^{l+u} max((1 - |f(\mathbf{x}_i)|), 0), \tag{2.6}$$

where the first term is the traditional hinge loss on the labeled samples, the second term refers to the regularization constraint to avoid over-fitting. The Third term, named as the hat loss, is introduced on the unlabeled set. The comparison of the loss functions on labeled and unlabeled samples are illustrated in Figure 2.5. As observed from the figure, a huge penalty is imposed on the unlabeled samples lying between the ± 1 margins so as to move the decision boundary towards the low-density regions.

The objective of TSVM is difficult to optimize as the hat loss is non-convex. Researchers have proposed many approaches for the optimization. A typical approach, proposed by Collobert et al. [Collobert et al., 2006], decomposes the hat loss as a summation of a convex and a concave function. Using the Concave-Convex Procedure (CCCP), the objective can be optimized with ease.

2.3.4 Laplacian Support Vector Machine

LapSVM [Melacci and Belkin, 2011], short for Laplacian Support Vector Machine, falls into the general category of Graph-based SSL [Zhu et al., 2003a], which represents the samples as nodes on a conceptual graph. Graph-based SSL usually relies on the smoothness assumption that nodes lying closely in the sample space are likely to share the same

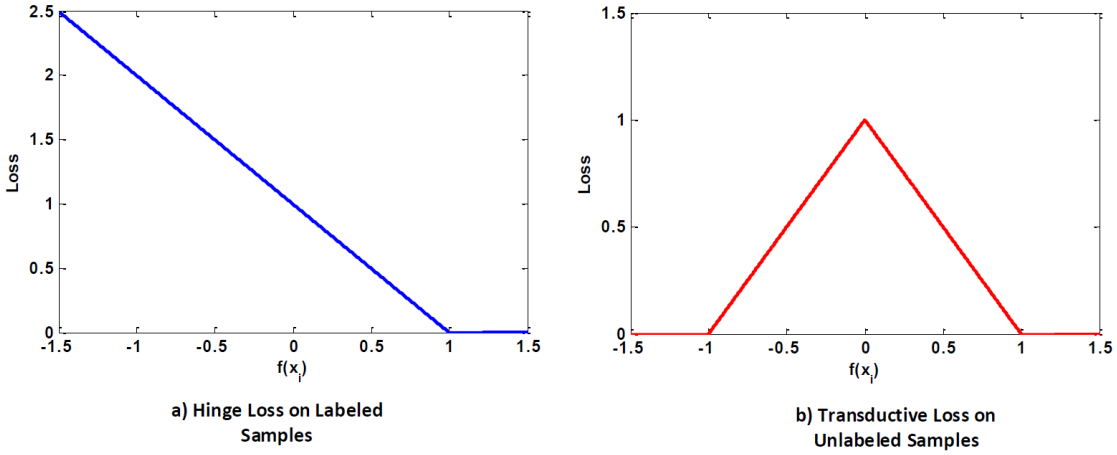

Figure 2.5: Comparison of losses on labeled and unlabeled samples. The left figure shows the hinge loss imposed on labeled samples. The right figure shows the hat loss imposed on unlabeled samples. Clearly, the hinge loss is convex, while the hat loss is non-convex.

label. This assumption yields the similar preference to TSVM, and usually leads to a decision boundary through low-density regions. A typical formulation of LapSVM is defined as

$$\mathcal{J} = \sum_{i=1}^{l} max(1 - y_i f(\mathbf{x}_i), 0) + \gamma_A ||f||_A^2 + \gamma_I \sum_{i,j} s_{ij} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2, \tag{2.7}$$

where s_{ij} refers to the similarity between sample $\mathbf{x}_i$ and $\mathbf{x}_j$, and is usually formulated as

$$s_{ij} = exp(-||\mathbf{x}_i - \mathbf{x}_j||^2 / 2\sigma^2). \tag{2.8}$$

In Eqn. (2.8), σ is the hyper-parameter for distance measurement. Similar to Eqn. (2.6), the second term refers to the regularization constraint. The last term corresponds to the smoothness assumption penalizing on the difference of the predictions between samples with small distance. By optimizing Eqn. (2.7), the labeled samples gradually propagate their labels to corresponding near-by nodes in the graph.

Both TSVM and LapSVM are taken as the basic components of the proposed Adaptive Learning framework in Chapter 5. More details can be found in the corresponding

sections.

Chapter 3

Part-based Deep Facial Representation

Multi-modality usually leads to significant intra-class variations. The complex combination of such variations is highly dependent on the photographic conditions in a specific scenario, and thus is hard to predict in advance. Moreover, the interaction among different modalities poses great challenges to feature learning methods with shallow structure. To solve these issues, the following two chapters aim to tackle the multi-modal challenge as described in Chapter 1 with deep learning based approaches. In this chapter, we propose to learn a part-based deep representation in addressing face verification in the wild. The task of face verification is to distinguish whether two face images belong to the same individual. It has long been an active research problem of computer vision. In particular, face verification under unconstrained settings has attracted much research attention in recent years. The release of several public data sets, e.g., Youtube Faces dataset (YTF) [Wolf et al., 2011] and Labeled Faces in the Wild dataset (LFW) [Huang et al., 2007b], has greatly boosted the development of face verification techniques.

Unconstrained photographic conditions bring about various challenges to face verification in the wild. Among them, one prominent challenge is the severe local distortions, such as pose variations, illumination changes and different facial expressions. To solve this issue, many state-of-the-art approaches for face verification [Simonyan et al., 2013, Cui et al., 2013, Li et al., 2013] are built on part-based face representation to take advantages of local representation robustness to local distortions. However, most part-based approaches are built on hand-crafted features, such as Local Binary Pattern (LBP) [Ojala et al., 1996], Scale-Invariant Feature Transform (SIFT) [Lowe, 2004], and Gabor features [Liu and Wechsler, 2002]. Those generic features are not specifically designed for the face verification tasks, and thus suffer from following issues. Firstly of all, some characteristic visual information may be lost in the extraction (especially their quantization) stage, which unfortunately cannot be recovered in later stages. Such information lost may severely damage the face verification performance. Moreover, another weakness of those hand-crafted features is the high requirement on the accuracy of facial alignment. Face alignment alone is considered to be quite challenging for face images captured in the wild. These issues become even more complicated if various combinations of different features, alignment methods and learning algorithms are considered for choice.

Recently, the well developed deep learning methods propose to solve the above issues by learning the feature representation and classifiers jointly for a specific task, and see great success for various computer vision tasks [Krizhevsky et al., 2012, Farabet et al., 2013, Huang et al., 2012b, Sun et al., 2013a, Nair and Hinton, 2010]. Within a general deep neural network, the bottom layers usually extract elementary visual features, e.g., edges or corners, and feed forward the output to the higher layers which then extract higher-level features, such as object parts. The features extracted by the network are optimized in a supervised manner to fit a specified task and bring significant performance boosting. Inspired by the impressive performances, we also propose a deep learning

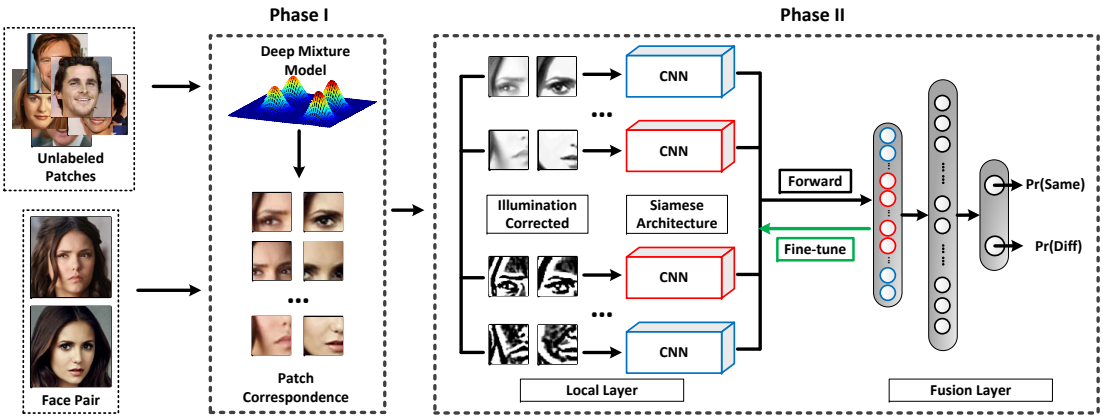

Figure 3.1: Flowchart of the proposed framework. A deep mixture model (DMM) is firstly trained with unlabeld local patches to capture the spatial and appearance distribution over faces. For each image pair, a pair of local patches is acquired for each mixture component in DMM with regard to the corresponding responses. The selected patch pairs are then pre-processed with several illumination correction methods and fed into multiple sub-CNNs for supervised pre-training. The pre-trained sub-CNNs are finally fused together with a holistic fusion layer.

method to solve face verification problem in this work. Although for face verification the part-based approaches have been proven effective with hand-crafted features [Li et al., 2013, Simonyan et al., 2013], the power of part based model may be weakened by the improper hand-crafted features as aforementioned. Therefore, how to learn a suitable local feature representation is a critical problem for face verification, which however has not been explored much yet. Most of the existing deep learning networks [Sun et al., 2014b, Sun et al., 2014a, Taigman et al., 2014] aim to learn global features from the full face images, instead of robust local ones as advocated in this chapter. Moreover, most aforementioned works are built on well-aligned faces, while approaches for verifying faces with natural mis-alignment are still rare.

In this work, we introduce a novel two-stage deep model to automatically learn robust local face representations for face verification in the wild. In contrast to previous works, our proposed model does not require the faces to be well aligned, and deals

with the more realistic wild setting where there exists considerable mis-alignment of faces. This makes our proposed model more appealing for practical applications. The proposed deep model automatically matches the local face patches via a novel Deep Mixture Model (DMM), and then adopts Convolutional Fusion Network (CFN) to learn a part-based face representation as shown in Figure 3.1. Benefited from these two stages, the output face representations are more robust to local variations in terms of pose, illumination, etc.

More concretely, the first layer of CFN (local layer) is pre-trained on local patches of different scales, geometric positions and illuminations. The following layer (holistic layer) learns a fully-connected classifier built on the local responses forwarded from the local layer. Conventional CNN assumes that the feature distribution is uniform over the face, thus extracts features with the same convolutional kernels for different face regions. This assumption usually does not hold in practice. In contrast, our network models the non-stationary feature distribution explicitly. Each sub-CNN in the local layer captures features that are specific for patches in the given face regions with the given illumination. Such composite structure leads to representation with tolerance to local distortions, and meanwhile captures the holistic information with the global fusion.

The problem of large pose variations is further addressed via exploring the semantic patch correspondence. Recent works [Li et al., 2013, Wright and Hua, 2009] indicate that semantically normalized patches usually improve the performance for face matching problems with various poses. In this chapter, a mixture model of deep representation is proposed to acquire the patch correspondence. Different from previous approaches relying on manually designed features, both the representation and the mixture component parameters are optimized together by maximizing the posterior probability of the model. With the deep mixture model, patches of highest responses to the same component are taken as matched for each pair of images. The matched pairs are further

ranked in terms of their discriminative scores, and those top ranked patches are chosen as the inputs for CFN. The screening process results in higher efficiency of the proposed network while retaining the verification performance.

In general, our contributions of this chapter can be summarized as follows.

1. We propose a novel way of learning a part-based face representation with Convolutional Fusion Network built on multiple CNN models. Different representations are learnt for different facial regions to adapt to the geometrically non-stationary distribution. The independence leads to better generalization performance with the holistic fusion.

2. We propose a Deep Mixture Model to obtain the semantic correspondence of patches to handle pose variation. Within the DMM network, the mixture components and the representation are jointly optimized, which is proven to be effective by extensive experiments.

3. We propose a new patch selection procedure to maintain only the discriminative patches for face verification. Such selection largely reduces the number of patches needed in CFN and leads to considerate improvement of accuracy over manually selected approach.

The proposed network is evaluated on two benchmark databases for face verification – YTF and LFW, and achieves competitive results with the state-of-the-arts.

3.1 Related Work

Among a large number of topics related, we list two aspects of research that are most relevant to our methods in this chapter.

3.1.1 Part-based Representation for Face Images

Face related tasks have attracted considerable attention due to their application potential. Seeking for a good representation of face images has long been an interesting topic for researchers.

Many methods on face representation [Tan and Triggs, 2010,Ahonen et al., 2006,Hussain et al., 2012] have been proposed during the past few decades. These methods can be roughly categorized into holistic and local approaches. Classic works on holistic features, such as Principal Component Analysis [Turk and Pentland, 1991], are mainly subspace-based approaches that try to represent face images with the subspace basis. Compared with holistic features, local features are more robust and stable to local changes and have been widely used recently. Gabor [Liu and Wechsler, 2002], Local Binary Pattern (LBP) [Ojala et al., 1996] and Bag of Words (BoW) [Sikka et al., 2012] features are classic representations capturing the local information. Gabor feature captures the spatial-frequency information and is found to be robust to the illumination variation. LBP captures contrast information for each pixel by referring to its neighboring points. BoW represents the image as an orderless collection of local features extracted in densely sampled patches.

Part-based face representation [Kim et al., 2003, Kim et al., 2005] is a popular way of capturing the local information and has been successfully applied to facial expression recognition [Sikka et al., 2012, Zhao et al., 2013], face parsing [Luo et al., 2012], face identification [Zhu et al., 2012] and face verification [Simonyan et al., 2013]. Karan et al. [Sikka et al., 2012] proposed a BoW representation of face images for facial expression recognition. They extracted SIFT descriptors on densely sampled patches of multi-scale and then built the codebook. Luo et al. [Luo et al., 2012] introduced a hierarchical face parser. The parser combines the results of part detectors and component detectors to transform the face image into a label map. Zhu et al. [Zhu et al., 2012] targeted at face

recognition problems with a small number of training samples. They conducted collaborative representation based classification on the face patches and combined the results of all the multi-scale patches.

There have also been some recent works with part-based representation on face verification, which refreshed the state-of-the-art performance, especially for unconstrained face verification in the wild. To name a few, Li et al. [Li et al., 2013] built a Gaussian Mixture Model (GMM) in terms of both appearance and spatial information to discover the correspondence between the patches in pair. The model is trained with LBP and SIFT features extracted from densely sampled patches. Their approach improved the state-of-the-art performance by around 4% on LFW with the most strict setting. In [Simonyan et al., 2013], Fisher Vector (FV), a typical descriptor for object recognition, was applied on LFW, and improved the performance further. FV in their work is built on SIFT feature extracted from the patches scanned densely through the images.

The aforementioned methods extract the same features from the different facial parts. However, we consider the feature distribution is not stationary over the whole face in this chapter, i.e., the learnt filters are different for different face regions. Without the hand-crafted features as in mentioned works, the proposed fusion network learns the feature representation automatically with direct guidance of the facial identities.

3.1.2 Deep Learning

The breakthrough by Hinton and Salakhutdinov [Hinton and Salakhutdinov, 2006] triggered the enthusiasm for deep learning in both academia and industry. By stacking multiple non-linear layers, deep neural networks are able to extract more abstract features automatically than the hand-crafted features.

Over the past few years, such a deep structure has been successfully applied in many computer vision fields [Krizhevsky et al., 2012, Farabet et al., 2013, Huang et al.,

2012b, Nair and Hinton, 2010, Sun et al., 2013a]. To name just a few, Krizhevsky et al. [Krizhevsky et al., 2012] won the ImageNet contest in 2012 by training deep CNNs fine-tuned with multiple GPUs. Sun et al. [Sun et al., 2013a] proposed a three-level cascade of convolutional networks for facial keypoints detection and outperformed the state-of-the-art methods in both detection accuracy and reliability. Ouyang and Wang [Ouyang and Wang, 2013] proposed joint deep learning framework to address pedestrian detection. Feature extraction, deformation handling and occlusion handling are incorporated in a unified framework and achieves the best performance on the Caltech dataset.

Several recent works also apply deep learning to face verification task. Huang et al. [Huang et al., 2012b] developed a convolutional Restricted Boltzman Machine (RBM) and evaluated it on the LFW-a database (with face alignment). The proposed method achieves comparable result to those with hand-crafted features. Chopra et al. [Chopra et al., 2005] defined a mapping from input space to the target space to approximate the semantic distance in the original space. The mapping is learned with two symmetric neural networks that share the same weights to tackle face verification problem. Liao et al. [Liao et al., 2013] proposed a three-layered hierarchy without explicit detection and alignment stages in testing. However, these networks are trained with full face images only and do not specifically handle local variations. Different from the aforementioned papers, our network learns a composite representation from both the holistic faces and local patches by integrating the responses of discriminative local sub-nets.

A gradual increase in the amount of data significantly improves the verification accuracy of deep models. Sun et al. [Sun et al., 2014b] learnt a set of high-level features through a multi-class identification task. The network is trained on pre-defined face patches based on the landmark positions. The performance is further improved by Sun et al. [Sun et al., 2014a], in which the network is trained by jointly optimizing the identification and verification objectives. Taigman et al. [Taigman et al., 2014] introduced

the largest facial dataset to-date, which is used to learn an effective representation. The learnt presentation is directly applied on LFW and achieves close accuracy to that of human beings. The above deep networks are trained with an assumption that face images are well aligned. In contrast, the proposed framework is learnt with the existence of mis-alignment. To handle such mis-alignment, a deep mixture model network is proposed to capture the spatial-appearance distribution over faces. The DMM network automatically retrieves the patch correspondence, which is proven to be effective for unconstrained face verification.

3.2 Convolutional Fusion Network

Most state-of-the-art approaches evaluated on benchmark datasets for face verification are built on hand-crafted features [Li et al., 2013, Simonyan et al., 2013, Hu et al., 2013]. Instead, we address the problem by learning a part-based face representation automatically with Convolutional Neural Network (CNN). Conventional CNN is built by stacking multiple convolutional layers and pooling layers. The cascade of convolution-pooling structure provides certain robustness to shifting and rotation variations. However, the final features capture the facial patterns in a holistic manner, i.e., standard CNN learns a holistic stationary distribution of features. In contrast, local representations are more robust to local facial distortions which are common in face images in the wild. Thus, we aim at designing a network capturing both holistic and local facial properties. Introducing local information to CNN enables the network to learn a more diverse and complex presentation and leads to potential improvement.

Accordingly, the proposed Convolutional Fusion Network, illustrated in Figure 3.1, has a structure of two layers – the local layer and the fusion layer. The local layer is composed of several parallel sub-CNNs corresponding to the local face patches (the full-face images are resized and treated the same as local patches) , and thus captures features

with regard to the local variations. The fusion layer contains a fully-connected layer followed by a softmax classifier. It integrates the local responses to acquire a holistic view of the original image. Sub-CNNs are pre-trained separately to guarantee a certain level of independence. Such independence leads to a mutual complementary interaction among sub-CNNs, resulting in a considerate improvement with fusion layer.

Illumination is also a significant factor degrading the performance of unconstrained face verification. Hua and Akbarzadeh [Hua and Akbarzadeh, 2009] included the illumination pre-processing step and reported a considerate performance improvement. In this chapter, the face images are pre-processed with several standard illumination correction methods, including Local Ternary pattern [Tan and Triggs, 2010], Self-Quotient Image [Wang et al., 2004] and Histogram Equalization [Gonzalez and Woods, 2002]. The local patch are then cropped from lighting-corrected images, and passed to corresponding sub-CNNs.

For the classification task, the final fusion layer is a fully-connected layer. We denote the output of sub-CNN i as $h^{(i)}(\cdot)$, and the forward propagation of the final fusion layer can be represented as

$$\mathbf{y}' = \sum_{i=1}^{N} \mathbf{W}_f^{(i)} \cdot h^{(i)}(\cdot) + b_f, \tag{3.1}$$

where $\mathbf{W}_f^{(i)}$ and b_f are the corresponding weights and bias in the fusion layer, and N is the number of sub-CNNs. As we are tackling the problem of face verification, the output $\mathbf{y}'$ is a 2x1 vector, in which the i-th entry represents the possibility that the given sample should be classified as the i-th class.

3.2.1 Siamese Architecture

Each sub-CNN in CFN has a composite structure of two identical sub-networks as illustrated in Figure 3.2. Such a structure is termed as Siamese Architecture in [Chopra et al.,

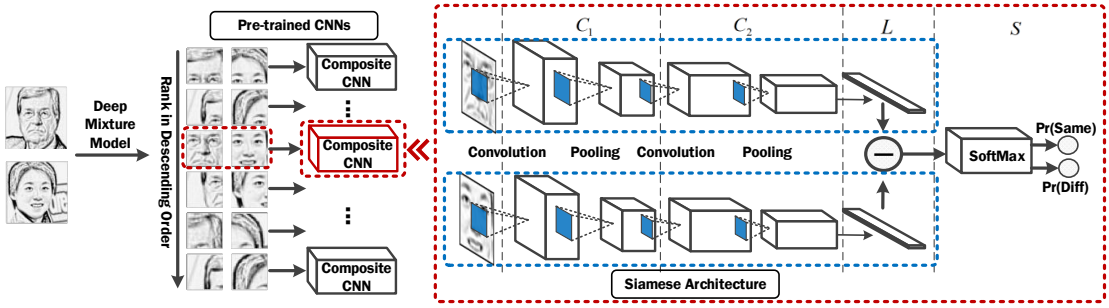

Figure 3.2: Siamese architecture. Each sub-CNN corresponding to a local support patch is composed of two identical CNNs that share the same weights. Such identical CNNs define a mapping from the input space to a space for a better similarity measurement.

2005, Nair and Hinton, 2010]. The two networks share the same weights, and define a mapping from the input feature space to a low-dimensional space where faces are close in terms of ℓ_1 distance if they are of the same identity.

Each sub-network in the composite structure is a Convolutional Neural Network (CNN), for which we follow the standard configuration in [Krizhevsky et al., 2012]. Each CNN contains two convolution layers C_1 and C_2, each of which is followed by a max-pooling layer. The output of convolutional layer is passed through a non-linear activation function before being forwarded to the pooling layer. In our networks, we use Rectified Linear unit (ReLu). Accordingly, the forward function can be represented as

$$h(\mathbf{x}_i) = max(0, \mathbf{W}_c^T \mathbf{x}_i + b_c), \tag{3.2}$$

where $\mathbf{W}_c$ and b_c represent the weight and bias of the corresponding convolutional layer. The last layer before softmax is a mapping layer L consisting of two fully-connected linear layers. The output of this linear layer is the final representation for each face pair and can be computed as

$$L(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}) = ||g(\mathbf{x}_i^{(1)}) - g(\mathbf{x}_i^{(2)})||_1, \tag{3.3}$$

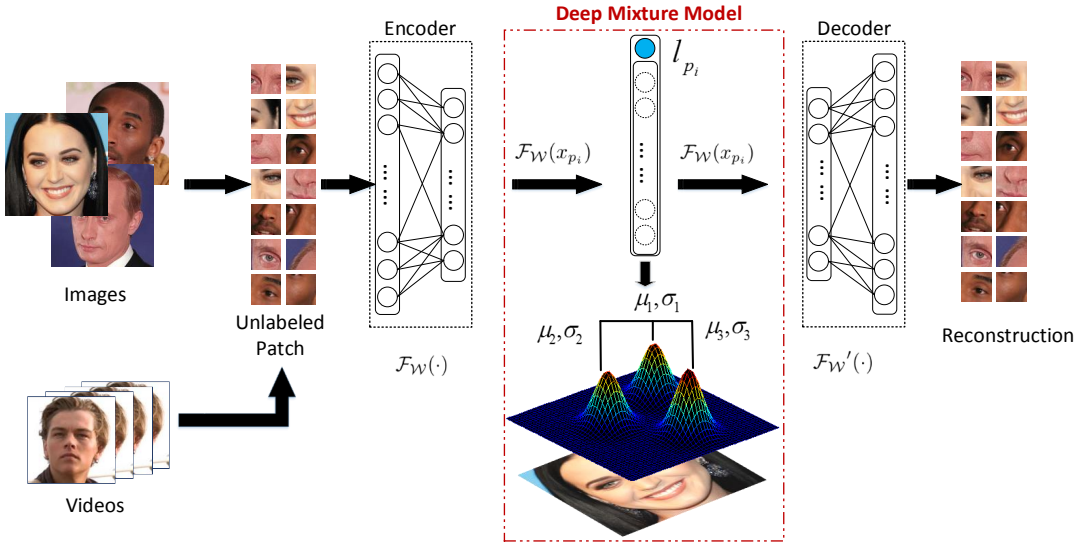

Figure 3.3: DMM network structure. The proposed network is of an encoder-decoder structure similar to Autoencoder and is trained with unlabeled patches extracted from input images or videos. The encoded features are augmented with the corresponding location vectors and applied to train the mixture model. The mixture component and the encoding function are jointly learnt within the unified framework.

where $g(\cdot)$ represents the mapping from the input space to the final feature space, and $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ are the two faces in a pair. By taking the ℓ_1 norm, $L(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})$ becomes a scalar variable.

The output of $L(\mathbf{x}_i)$ is finally forwarded to a softmax layer. As a binary classification problem, the learnable weight is a two column vector $\mathbf{W}_s = \{\mathbf{W}_s^{(1)}, \mathbf{W}_s^{(2)}\}$. Also, the softmax layer can be seen as a fully-connected layer with a weight matrix $\mathbf{W}$. The posterior probability of $\mathbf{x}_i$ labeled as y_i is

$$P(y = y_i | \mathbf{W}_s, b_s, \mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}) = \frac{exp(\mathbf{W}_s^{(y_i)} \cdot L(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}) + b_s)}{\sum_{j=1}^{2} exp(\mathbf{W}_s^{(j)} \cdot L(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}) + b_s)}. \tag{3.4}$$

For the verification task, $y_i \in \{0, 1\}$. Accordingly, the cost function is formulated as

follows

$$\mathcal{J}_{CFN} = -\sum_{i=1}^{n} \log P(y = y_i | \mathbf{W}_s, b_s, \mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}). \tag{3.5}$$

3.3 Pose-invariant Patch Selection

To acquire the local information, sub-CNNs of CFN are pre-trained on the discriminative facial parts, and thus the selection of patches will largely affect the performance. A typical part-based approach is built on patches that are densely sampled with overlap as in [Li et al., 2013, Simonyan et al., 2013]. Intuitively, we can generate patches following the same strategy. However, there are mainly two reasons prohibiting us from doing so. First, such an approach will generate a huge network with an unaffordable number of sub-CNNs since each local patch requires one sub-network in Figure 3.2. The unaffordable computation cost makes it infeasible to adopt this approach. Second, large networks are difficult to train even if we ignore the computation cost. With too many parameters to learn, it is hard for the network to converge. Moreover, the optimization of the deep network is non-convex, and thus sensitive to the initialization of parameters. It easily falls into the "basin" of poor local minimum without a proper initialization.

Another way to utilize local information is to extract patches with regard to the key facial landmarks, such as eyes, nose, mouth, etc. This kind of approaches largely relies on the precision of landmark detectors. However, the unconstrained photography conditions still remain challenging for most existing landmark detectors. Moreover, accurate landmark detectors usually demand a large set of outside training samples, which are not always available. Thus, this strategy is prohibited for some datasets in the wild, e.g., LFW under the most restricted condition.

Our approach is built on the assumption that the face images are captured in the wild and no accurate landmarks are available, and thus the faces are only roughly aligned.

The pose variation has proven to be an important factor impacting the face recognition accuracy. We propose to learn a Deep Mixture Model (DMM) to capture the spatial-appearance distribution over faces. By learning the mixture components, the correspondences of local patches are acquired to address the mis-matching brought by pose difference. Different from APEM [Li et al., 2013], our deep network learns both the representation for appearance and the mixture components jointly without reference to any manually designed features.

3.3.1 Deep Mixture Model

Given a set of unlabeled images, we divide each image into multiple overlapped grids. The image set then can be represented as a collection of local patches $\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N\}$. Each local patch $\mathbf{p}_i$ is represented as a spatial and appearance pair $[\mathbf{x}_{p_i}^T, \mathbf{l}_{p_i}^T]^T$, where $\mathbf{x}_{p_i}$ is the raw-pixel representation and $\mathbf{l}_{p_i}$ (each element within ranges from 0 to 1) is the normalized location vector .

Different from most existing works for learning a mixture model, our approach does not rely on hand-crafted features. Instead, the representation is learnt together with the mixture components. Similar to Auto-encoder, the DMM network contains an encoder and a decoder as shown in Figure 3.3. The encoder maps the high-dimension data to a low-dimension code, and the decoder recovers the original input from the compressed code. In this work, the encoder is of a two layered structure: 800 hidden units for the first layer and 200 hidden units for the second layer. The decoder has a symmetric structure to the encoder. Also, the encoder and decoder have "tied" weights, i.e. the weight matrix for the decoder layer is the transpose of that for the corresponding decoder layer. The "encoded" feature is forward into the third layer, i.e. the mixture layer. The mixture layer is composed of multiple branches, each of which corresponds to a mixture component. The output of each component sub-net is the probability of certain patch

committed to the corresponding component.

Assume the encoding function defined by the deep network is $\mathcal{F}(\cdot\,; \mathbf{W}_e, \mathbf{b}_e)$, where $\mathbf{W}_e$ and $\mathbf{b}_e$ stand for the encoding weight and bias. By augmenting the compressed code and the location vector, the combined spatial-appearance feature is represented as $\mathbf{X}_{p_i} = [\mathcal{F}(\mathbf{x}_{p_i}; \mathbf{W}_e, \mathbf{b}_e)^T, \mathbf{l}_{p_i}^T]^T$, which is then forwarded to the following mixture layer. We formulate the deep mixture model in terms of Gaussian components as follows,

$$P(\mathbf{X}_{p_i}|\boldsymbol{\theta}) = \sum_{j=1}^{C} \omega_j \cdot \mathcal{N}(\mathbf{X}_{p_i}|\boldsymbol{\mu}_j, \sigma_j), \tag{3.6}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\mu}_i, \sigma_i | i = 1, 2, ..., C\}$, and $\boldsymbol{\mu}_i$ and σ_i are the mean and variance of the i-th component. $\mathcal{N}(\cdot)$ represents a normal distribution for the component with corresponding mixture weight w_i.

The DMM network is optimized by minimizing the following cost function

$$\begin{aligned} \mathcal{J}_{DMM}(\mathbf{W}, \mathbf{b}, \boldsymbol{\theta}) \;=\; & -\sum_{i=1}^{N} ln\, P(\mathbf{X}_{p_i}|\boldsymbol{\theta}) \\ & -\sum_{i=1}^{N} ln\, \frac{\max_j \mathcal{N}(\mathbf{X}_{p_i}|\boldsymbol{\mu}_j, \sigma_j)}{\sum_{j=1}^{C} \mathcal{N}(\mathbf{X}_{p_i}|\boldsymbol{\mu}_j, \sigma_j)} \\ & +\sum_{i=1}^{N} \alpha ||\mathbf{x}_{p_i} - \mathbf{x}'_{p_i}||^2, \end{aligned} \tag{3.7}$$

where α is a parameter controlling the contribution scale of the third term, and $\mathbf{x}'_{p_i}$ is the reconstruction of $\mathbf{x}_{p_i}$ and is computed as

$$\mathbf{x}'_{p_i} = \mathcal{F}'(\mathcal{F}(\mathbf{x}_{p_i}\,;\, \mathbf{W}_e, \mathbf{b}_e)\,;\, \mathbf{W}'_e, \mathbf{b}'_e), \tag{3.8}$$

where $\mathcal{F}'(\cdot\,;\, \mathbf{W}'_e, \mathbf{b}'_e)$ is the decoding function with the corresponding decoder weight $\mathbf{W}'_e$ and bias $\mathbf{b}'_e$.

The cost in Eqn. 3.7 is defined based on considerations on the following three aspects. Same as the standard Gaussian Mixture Model, the first term is defined as the

log likelihood function. For the second term, the proposed DMM aims to regularize that the spatial-appearance components correspond to different semantic facial parts, such as eyes, nose, etc. In other words, the learnt mixture components are expected to follow a spatially scattering distribution. Therefore, we introduce the second term to constrain that each sample is only committed to one component and its contribution to other components are neglectable. It is also important to note that, in DMM, the encoding of patches is jointly optimized with the component parameters. Direct optimization with regard to the first and second terms will result in an undesired global minimum where both $\mathbf{W}_e$ and b_e are all zero for the encoder. Therefore, the third term is introduced to penalize the construction error such that the representations of face patches are not mapped into the undesirable all-zero space.

The mixture parameters are only present in the mixture layer, and thus are independent of the reconstruction error. Accordingly, $\boldsymbol{\mu_k}$ and σ_k can be updated directly as follows.

$$\begin{aligned}\frac{\partial \mathcal{J}_{DMM}}{\partial \boldsymbol{\mu}_k} = (&-\frac{w_k}{P(\mathbf{X}_{p_i}|\boldsymbol{\theta})} + \frac{1}{\sum_{j=1}^{C}\mathcal{N}(\mathbf{X}_{p_i}|\boldsymbol{\mu}_j,\sigma_j)} \\ &- \frac{\mathbb{1}_{j=k}}{\max_j \mathcal{N}(\mathbf{X}_{p_i}|\boldsymbol{\mu}_j,\sigma_j)}) \cdot \frac{\partial \mathcal{N}(\mathbf{X}_{p_i}|\boldsymbol{\mu}_k,\sigma_k)}{\partial \boldsymbol{\mu}_k},\end{aligned} \tag{3.9}$$

$$\begin{aligned}\frac{\partial \mathcal{J}_{DMM}}{\partial \sigma_k} = (&-\frac{w_k}{P(\mathbf{X}_{p_i}|\boldsymbol{\theta})} + \frac{1}{\sum_{j=1}^{C}\mathcal{N}(\mathbf{X}_{p_i}|\boldsymbol{\mu}_j,\sigma_j)} \\ &- \frac{\mathbb{1}_{j=k}}{\max_j \mathcal{N}(\mathbf{X}_{p_i}|\boldsymbol{\mu}_j,\sigma_j)}) \cdot \frac{\partial \mathcal{N}(\mathbf{X}_{p_i}|\boldsymbol{\mu}_k,\sigma_k)}{\partial \sigma_k}.\end{aligned} \tag{3.10}$$

The optimization of $\mathbf{W}$ and b can be conducted with the standard back-propagation algorithm.

3.3.2 Local Patch Matching

The acquired DMM reflects the distribution of spatial and appearance feature over the faces. By assigned each face patch to its "Nearest" mixture component, we are able to cluster the patches in terms of the encoded similarity. Within each face pair, face patches with the maximal responses to the same mixture component are considered as matched. Therefore, the number of components determines the number of sub-nets that need to be pre-trained. Large number of chosen patches will result in a huge computation cost. Instead, we assume that not all the patches will contribute to the final verification problem. Therefore, it is desirable to retain only those discriminative patches without impacting the generalized performance.

This task can be interpreted as a feature selection problem [Weston et al., 2000, Zhai et al., 2012], which selects a subset of features while preserving or even improving the discriminative ability of the classifier. Suppose we are given n training samples $\{(\mathbf{F}_1, y_1)$, ..., $(\mathbf{F}_n, y_n)\}$, where $\mathbf{F}_i \in D^d$ and $y_i \in \{-1, +1\}$ is the label of $\mathbf{F}_i$. For face verification, the training samples are given in pairs. The task is to tell whether or not the paired samples (probe and gallery) are of the same identity. We denote $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ as the first and second face in the i-th pair as in Eqn. (3.3). The input vector for the feature selection process is computed by $\mathbf{F}_i = |\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}|$, where $|\cdot|$ computes the element-wise absolute value.

In both [Zhai et al., 2012] and [Weston et al., 2000], an indicator vector $\delta = \{\delta_1, ..., \delta_d\} \in \{0, 1\}^d$ is introduced to define whether a certain feature dimension is selected, i.e. $\delta_j = 1$ indicates the j-th dimension is a "support feature". Instead of finding the pixel-wise discriminative features as in [Weston et al., 2000, Zhai et al., 2012], we aim to select the discriminative patches. With the learnt C-component DMM, each face pair $\mathbf{F}_i$ is represented as a concatenated vector $\mathbf{A}_i = \{\mathbf{p}_i^{(1)}, \mathbf{p}_i^{(2)}, ..., \mathbf{p}_i^{(C)}\}$. Note that $\mathbf{p}_i^{(j)}$ here represents the matched patch to component j in the difference vector $\mathbf{F}_i$. The specific

definition is given as follows,

$$\mathbf{p}_i^{(j)} = arg\,max_{\mathbf{p}_k}\ \mathcal{N}\left(\mathbf{X}_{p_k}|\boldsymbol{\mu}_j, \sigma_j\right) \quad \forall \mathbf{p}_k \in \mathbf{x}_i^{(\cdot)}. \tag{3.11}$$

A standard SVM classifier is applied directly on $\mathbf{A}_i$. The weight vector of SVM can be then decomposed as $\mathbf{W} = \{\mathbf{W}^{(1)}, ..., \mathbf{W}^{(C)}\}^T$ with regard to $\mathbf{A}_i$. In this work, we simplify the problem by approximating the indicator vector in [Weston et al., 2000, Zhai et al., 2012] as the weight vector. Correspondingly, the problem is transformed into a classic SVM issue. The classifier is

$$f(\mathbf{A}_i) = \mathbf{W}^T \mathbf{A}_i + b, \tag{3.12}$$

where b is the bias.

Note that a pixel in the original image may be included in multiple patches. By minimizing the ℓ_2 term $||\mathbf{W}||^2$ in the cost function, the corresponding duplicate pixels are assigned with the same weight if no individual normalization within each patch. Therefore, the discriminative scores of the duplicates in different patches are consistent. We define the discriminative score as the overall contribution of pixels within the patch to the decision boundary. The discriminative score $Sr^{(i)}$ of patch $\mathbf{p}^{(i)}$ is computed as

$$Sr^{(i)} = ||\mathbf{W}^{(i)}||_1. \tag{3.13}$$

Patches are then sorted in terms of the corresponding discriminative scores, and the top K patches are chosen as support patches.

Support patches tend to be those containing key facial components closely related to face identification, such as eyes and forehead. While, least informative patches include little information on either the outline of faces or key facial landmarks.

3.4 Training the Networks

The whole framework can be largely divided into two parts: 1) Deep Mixture Model to find the patch correspondence and 2) Convolutional Fusion Network for face verification. Both networks are large and hard to train directly without getting stuck at undesired local minimum. Erhan et al. [Erhan et al., 2010] mentioned that pre-training provides a prior knowledge that can help reduce the strong dependencies between parameters across layers and locates the network in a region within the parameter space, such that a better optimum is found for the training criterion. We include some details on the training strategies for both networks as follows.

DMM. An initial representation is essential to avoid undesired clustering performance for appearance-wise DMM. In this chapter, we follow the standard unsupervised pre-training method used for Auto-encoder. The network is pre-trained layer-by-layer with regard to the squared reconstruction error, i.e. the third term in Eqn. (3.7). For training the DMM network, we also need proper initialization for the location vectors. The location related part in $\boldsymbol{\mu}_i$ is initialized randomly with regard to a uniform distribution over $[0, 1]$. Moreover, for the starting 5 iterations, the encoder parameters ($\mathbf{W}_e$ and $\mathbf{b}_e$ in the 1st and 2nd layer) are not updated. In such a way, we acquire a proper geometric initialization for the mixture components.

CFN. Convolutional Fusion Network is initialized with the supervised pre-training. Selecting local patches can be viewed as a way of obtaining a good prior for the later fine-tuning stage. The pair of local patches shares the same label as the full-face pair, i.e. patches generated from the "matched" face pairs are also labeled as "matched". Therefore, each sub-CNN in the local layer can be pre-trained with the label information. After the supervised pre-training, the outputs of all the sub-CNNs are concatenated as a super-vector for each face instance, which is then fed forward to the fusion

layer. A universal fine-tuning is then applied with back propagation through the whole network. Experiments show that the final fusion stage results in a considerate performance improvement.

3.5 Experiments

The proposed network is aimed at face verification under the unconstrained conditions with variations on pose and illumination. Extensive experiments are conducted on two benchmark datasets for face verification in the wild – Youtube Faces dataset (YTF) and Labeled Faces in the Wild dataset (LFW). Examples of YTF and LFW is shown in Figure 3.4. The results are compared with several state-of-the-art approaches.

3.5.1 YouTube Faces Database

YTF is a dataset designed for studying the problem of unconstrained face verification in videos. It contains $3,425$ unconstrained videos of $1,595$ celebrities. In the standard protocol, the evaluation set is composed of $5,000$ pre-defined video pairs and is divided into 10 mutually exclusive folds. The average verification accuracy over the ten folds is reported for comparison.

3.5.1.1 Experiment Settings

We address the problem of verification of two face videos as the matching problem of two sets of frames. Specifically speaking, 20 frames are drawn randomly from each video within the pair to generate 20 frame pairs. The average matching score of the 20 frame pairs is taken as the matching score of the corresponding video pair. In the following experiments, we directly take the roughly aligned faces provided. Within each

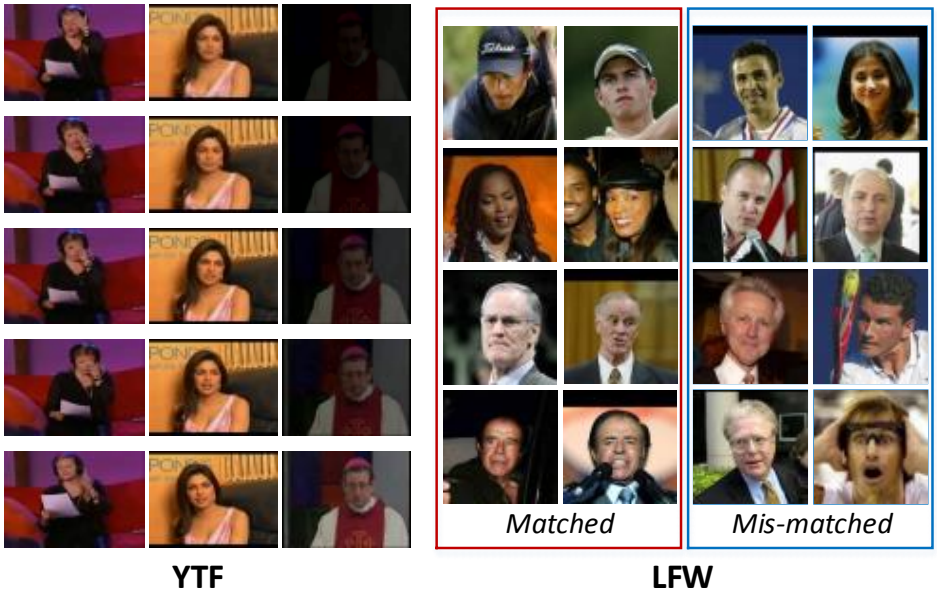

Figure 3.4: Examples from YTF (left) and LFW (right). Both datasets include variations on pose, illumination and facial expressions that has large influence on the matching performance. Moreover, occlusion, frame blur and scene transition, which are common in videos, make YTF even more challenging.

frame, the face is cropped from the center down-scaled by 2.2 and is of size 144×144. The face images are then processed with two common illumination correction methods – Histogram Equalization (HE) and Local Ternary Pattern (LTP) [Tan and Triggs, 2010]. For LTP, the gamma parameter is set as 0.2, and the sigma values for inner and outer Gaussian filter are set as 0.2 and 1, respectively. Together with RGB images, three copies of each images are adopted as inputs.

Pre-processed face images are scanned by sliding windows of size 40×40 and 60×60. The corresponding sliding strides are 20 and 30 pixels, respectively. Thus, we extract 44 local patches in each face image. These patches are resized to 32×32, and used as inputs of the DMM network.

CFN Structure. The whole network contains 18 sub-nets of Siamese Architecture in the local layer and a linear layer followed by a softmax layer in the fusion layer. Each sub-

Figure 3.5: Convolutional kernels computed. Each block corresponds to a selected patch with its learnt convolutional kernels in the first layer. Clearly, the learnt kernels are different for different facial patches.

network i has a four-layer structure consisting of two convolutional layers $C_1^{(i)}$ and $C_2^{(i)}$, one linear layer $L^{(i)}$ and one softmax layer $S^{(i)}$. $C_1^{(i)}$ contains 40 convolutional kernels with size 7×7, and $C_2^{(i)}$ has 40 kernels of size 5×5, and $L^{(i)}$ has 100 neurons. Both convolutional layers are followed by max-pooling of shape 2×2 with pooling stride 2×2.

Examples of learnt convolution kernels are shown in Figure 3.5. The convolutional kernels are learnt to reflect the discriminative information for the given local regions. For patches with complex facial structure (Full Face and patch 2), there are more high frequency kernels. While, for less complex patches (Patch 3, 4 and 5), the learnt kernels are mostly edge-like filters.

To further reduce over-fitting, drop-out [Hinton et al., 2012] is applied on each layer of sub-CNNs, except for the softmax layer. The drop-out rate is 0.2 for convolutional layers $C_1^{(i)}$, $C_2^{(i)}$ and the linear layers $L^{(i)}$. We also include random noises in the input images, and the corruption probability of a single pixel is 0.1.

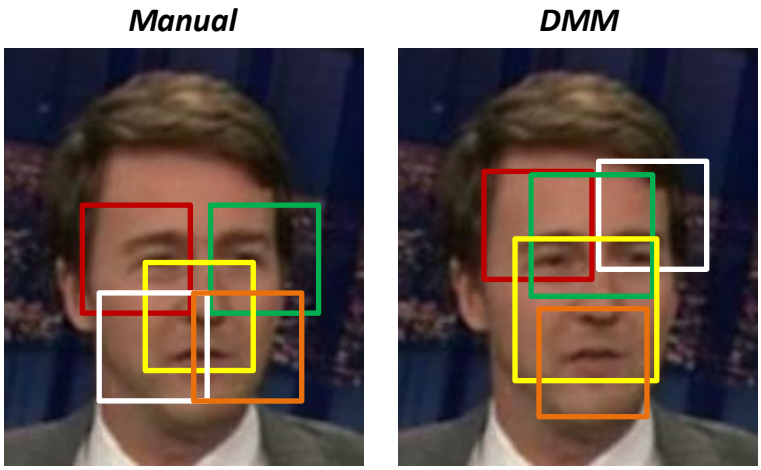

Figure 3.6: Illustration on manual patches (Left) and DMM patches (Right). Since faces are aligned roughly, we extract patches around eyes, nose and mouth corners with fixed locations. For DMM, the locations are learnt automatically w.r.t the spatial-appearance distribution. Compared with manual approach, DMM demonstrates a better tolerance to pose changes.

3.5.1.2 Comparison with the State-of-the-arts

The proposed approach, i.e. *DMM+CFN(3)*, is compared with several existing works reported on YTF in Table 3.1. Moreover, we include four variants of our method for self comparison.

CNN_Single shows the result of single CNN trained only with the full face images. *CFN_Manual* includes the local information by fusing local CNNs trained with manually selected patches. The patches are chosen intuitively around eyes, nose and mouth corners as shown in Figure 3.6. Comparison between *CNN_Single* and *CFN_Manual* indicates that the local information can bring considerate improvements (1.3% in our experiments) over holistic only approach. *DMM+CNN_Average* simply averages over pre-trained local CNNs. Different from *CFN_Manual*, local CNNs in this methods are

Methods	Acc. $\pm$ Err.(%)
MBGS L2 mean, LBP [Wolf et al., 2011]	76.4 $\pm$ 1.8
MBGS+SVM [Wolf and Levy, 2013]	78.9 $\pm$ 1.9
APEM-FUSION [Li et al., 2013]	79.1 $\pm$ 1.5
STFRD+PMML [Cui et al., 2013]	79.5 $\pm$ 2.5
VSOF+OSS [Mendez-Vazquez et al., 2013]	79.7 $\pm$ 1.8
DDML (LBP) [Hu et al., 2013]	81.3 $\pm$ 1.6
DDML (combined) [Hu et al., 2013]	**82.3** $\pm$ 1.5
CNN_Single	78.3 $\pm$ 1.4
CFN_Manual	79.6 $\pm$ 1.2
DMM+CNN_Average	79.5 $\pm$ 1.2
DMM+CFN (1)	80.9 $\pm$ 0.9
DMM+CFN (3)	**82.8** $\pm$ 0.9

Table 3.1: Comparison of mean accuracy and standard variance on YouTube Faces Database. The best performance is illustrated in bold.

learnt from patches acquired with the deep mixture model. As shown in the table, such simple approach can achieve almost the same performance as *CFN_Manual*. The performance is further improved by including the fusion stage into the learning process. *DMM+CFN(1)* is conducted on the images with only histogram equalization and improves *DMM+CNN_Average* by 1.4%. Fusion of more models is shown to be effective. The images used in *DMM+CFN(3)* are pre-processed with HE and LTP, respectively. Together with the original RGB images, the fusion model improves over single illumination based method *DMM+CFN(1)* by 1.9%.

Comparing with the state-of-the-art method on YTF – *DDML (combined)*, our approach improves the performance by 0.5%. *DDML (combined)* is also based on deep learning, but the networks learn a Mahalanobis distance metric from the hand-crafted features (LBP, DSIFT and SSIFT). However, our fusion network is directly learnt on the raw-pixel images.

The ROC curve is illustrated in Figure 3.7. Consistent with the comparisons in Table 3.1, our approach outperforms the existing methods reported on YTF.

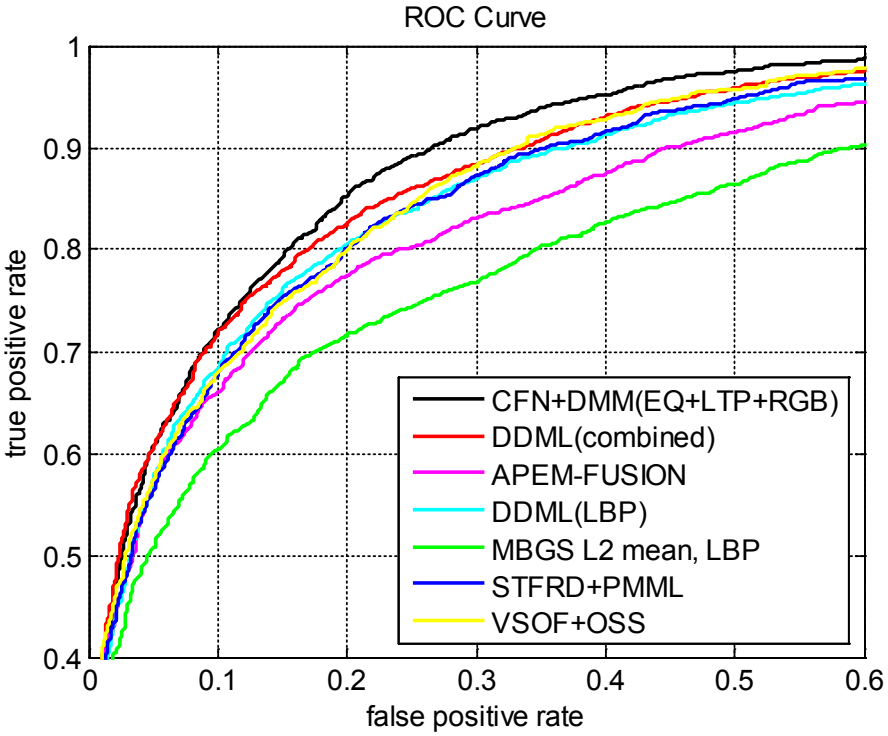

Figure 3.7: Comparison of ROC curves with the state-of-the-arts on YouTube Faces Database.

Here we also list some of the latest results published after the submission of our publication [Xiong et al., 2015a]. Li et al. [Li et al., 2015] proposed the Eigen-PEP model for video face recognition, and achieved 85.04 ± 1.49 on YTF and 88.97 ± 1.32 on LFW. In [Li et al., 2015], the performance is largely improved by including flipped frames and corrected labels, which are not used in our method. The accuracy without flipping is 82.40 ± 1.7, which is close to our results. Hu et al. [Hu et al., 2015] learnt the distance metrics form multiple features and achieved 81.28 ± 1.17 on YTF. Lu et al. [Lu et al., 2015] applied a reconstruction criterion to metric learning and achieved 81.86 ± 1.55.

3.5.2 Labeled Face in the Wild

LFW is a standard database collected to evaluate benchmark algorithms for face verification. It contains 13,000 images of 5,749 individuals downloaded from the Internet. LFW has the similar evaluation protocols as YTF: 6,000 pre-defined image pairs are divided into ten mutually exclusive folds and the average precision is reported.

3.5.2.1 Experiment Settings

In this paper, the experiments are conducted in the image-restricted scenario, i.e. only the given 6,000 pairs are allowed for training. We follow the most strict setting, i.e. no outside training data are used, even for landmark detection. The face images are only roughly aligned with an unsupervised method – deep funnel [Huang et al., 2012a]. We crop the central 144×144 region from the full-face image. DMM follows the same patch extraction strategy as that used for YTF.

Three general approaches of illumination correction are applied – Self-Quotient Image (SQI) [32], Local Ternary Pattern (LTP) [13] and Histogram Equalization (HE) . In SQI, the images are filtered with 7x7 Gaussian filter with bandwidth set as 2 and then normalized. The parameters for LTP are the same as those in YTF.

CFN Structure. The local networks are also of four layered structure – 20 convolutional kernels in $C_1^{(i)}$, 40 kernels in $C_2^{(i)}$, 100 hidden units in $L^{(i)}$ and a Softmax layer $S^{(i)}$. For LFW, we select the top-6 patches, and thus the final CFN is composed of 21 CNNs in the local layer.

3.5.2.2 Comparison with the State of the Arts

In this subsection, our approach is compared with some existing methods with the same setting, i.e. the image-restricted setting without outside training data. The only excep-

tion is NReLu [Nair and Hinton, 2010], in which face images are well-aligned and outside data are used for unsupervised pre-training. This approach built a DBN of siamese architecture, and thus is closely related to our method.

Table 3.2 shows the results of five different settings related to the proposed network. The number after each setting indicates the number of illumination correction methods included – for the 2-correction case images are pre-processed with only SQI and LTP. *CNN_Single(2)* reports the result of training CNNs only on the full-face images. Under this scenario, the fusion network only has two sub-CNNs on the full-face images after SQI and LTP respectively. The accuracy outperforms that of *NReLu* without unsupervised pre-training, and is comparable to their best performance with unsupervised pre-training based on outside unlabeled data. *DMM+CNN_Average(2)* simply averages over the confidence scores returned by pre-trained sub-CNNs. Performance with such a setting is even comparable with *APEM (fusion)* – only 0.1% difference. Further improvement is achieved by holistic back-propagation over the whole network, as shown by *DMM+CFN(2)*. The increase on mean accuracy is 1.55%, and can be up to 2.6% for some folds. The best results are achieved by fusion with all three illumination correction methods as shown for *DMM+CFN(3)*.

APEM [Li et al., 2013] is also based on selection of patches, and our method surpasses *APEM* with a single feature, either SIFT or LBP, by around 3.6%. The advance over APEM with feature fusion is 1.52%. There is a gap of 1.9% between fisher vector [Simonyan et al., 2013] and our method alone. However, by simply averaging with the results of APEM – *CFN+APEM*, we achieve the accuracy of *Fisher Vector*. The improvement by simply averaging with APEM demonstrates the features learnt in our fusion network is different from the hand-crafted features. Note that both *APEM* and *Fisher Vector* are built on images of large size (100×100 in APEM and 160×125 in FV), while our fusion network is only trained on images of small size 32×32.

Methods	Acc. ± Err.(%)
NReLu [Nair and Hinton, 2010]	80.73 ± 1.34
NReLu without Outside Data [Nair and Hinton, 2010]	79.25 ± 1.73
Hybrid descriptor-based [Wolf et al., 2008]	78.47 ± 0.51
V1/MKL [Pinto et al., 2009]	79.35 ± 0.51
APEM(LBP) [Li et al., 2013]	81.97 ± 1.90
APEM(SIFT) [Li et al., 2013]	81.88 ± 0.94
APEM(fusion) [Li et al., 2013]	**84.08** ± 1.2
Fisher Vector [Simonyan et al., 2013]	**87.47** ± 1.49
CNN_Single(2)	80.59 ± 1.54
CFN_Manual(2)	82.05 ± 1.6
DMM+CNN_Average(2)	83.93 ± 1.75
DMM+CFN (2)	85.48 ± 1.64
DMM+CFN (3)	**85.60** ± 1.67
CFN+APEM	**87.50** ± 1.57

Table 3.2: Comparison of mean accuracy and standard variance on Labeled Face in the Wild. The best performance is illustrated in bold.

Patch #	Full-face Included			Without Full-face		
	SQI	LTP	Combined	SQI	LTP	Combined
0	80.11 ± 1.73	81.07 ± 1.01	82.45 ± 1.40	-	-	-
1	81.67 ± 1.24	83.14 ± 1.61	84.48 ± 1.42	78.18 ± 1.54	77.92 ± 2.48	80.10 ± 2.10
2	83.25 ± 1.75	83.55 ± 1.49	85.18 ± 1.90	82.37 ± 2.27	81.95 ± 2011	84.35 ± 2.26
3	83.24 ± 1.72	83.67 ± 1.65	84.92 ± 1.72	82.33 ± 1.67	82.20 ± 2.02	83.98 ± 1.73
4	83.09 ± 1.94	83.7 ± 1.76	85.15 ± 1.46	82.27 ± 1.92	82.60 ± 2.33	84.18 ± 2.68
5	**83.34** ± 1.89	83.74 ± 1.76	85.24 ± 1.46	**82.38** ± 1.93	**83.10** ± 2.35	**84.50** ± 2.40
6	83.21 ± 1.95	**83.74** ± 1.69	**85.48** ± 1.64	82.10 ± 2.30	82.43 ± 2.43	84.20 ± 2.12

Table 3.3: Fusion Results. In each experiment set, results are reported by varying the number of local patches included. 0 means only the full-face images are used for training.

The ROC curve in Figure 3.8 illustrates the average performance over 10 folds. It is clear that our method outperforms APEM significantly and achieves a comparable performance with *Fisher Vector*.

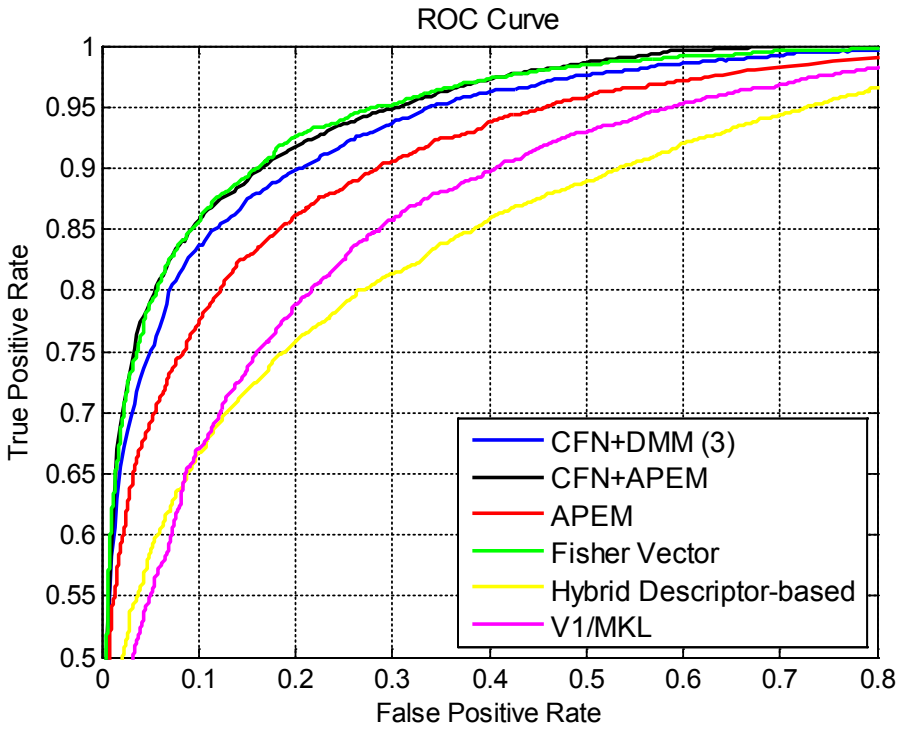

Figure 3.8: Comparison of ROC curves with the state-of-the-arts on the most strict setting of Labeled Face in the Wild.

3.5.2.3 Fusion Result Analysis

We conduct two sets of experiments to analyze the effect of several factors on fusion. The first set fuses the local patches with the full-face images. The second set studies the fusion among only the local patches. For each experiment set, we include three groups tested on the images after SQI, images after LTP and images after both SQI and LTP (*Combined* in Table 3.3), respectively. We also examine the influence of local patches in fusion by varying the number of patches included. These patches are added in the descending order with regard to their confidence scores defined by Eqn. 3.13.

Referring to the results in Table 3.3, sub-CNNs trained with full-face images have a considerate influence in fusion. Fusion with full-face images outperforms fusion with

only local patches by approximately 1.1%. Note that the local patches also demonstrate great influence. Generally, more local patches lead to higher accuracy in both experiment sets. As more patches are included, the performance gradually saturates. Fusing different pre-processing methods also contributes to the final fusion performance, and the increase on accuracy is around 1%.

3.5.3 Computation Analysis

The proposed framework can be divided into two parts – DMM and CFN. Both networks are implemented based on Theano[1] and Pylearn2[2]. All experiments are conducted on a single-core computer with GeForce GTX TITAN Black GPU. For both data sets, we extract 44 local patches from each face image, and random sample 60,000 patches for YTF and 45,000 patches for LFW as the inputs for DMM, respectively. In YTF, the training set of CFN includes 4,500 video pairs. Within each video pair, 20 frame pairs are randomly chosen. Accordingly, DMM takes 45s per iteration in training and CFN takes 33s per iteration for each sub-net. In LFW, the training set includes 5,400 image pairs for CFN. We also include random shifting, scaling and rotation to increase the diversity and scale of the training samples. As a result, the network is trained with 21,600 image pairs in total. Accordingly, DMM takes 36s per iteration in training and CFN takes 9s per iteration for each sub-nets. For faster computation, we can fix the convolution layers in the sub-nets of CFN, and only fine-tune the later fully-connected layers as many previous papers did. The corresponding results are only slightly degraded. The reported results are derived by setting the maximal training iteration number as 160 for DMM and 120 for CFN, respectively.

[1] http://deeplearning.net/software/theano/
[2] http://deeplearning.net/software/pylearn2/

3.6 Conclusions

In this Chapter, we propose a part-based learning scheme for face verification in the wild by introducing Convolutional Fusion Network. We fuse multiple sub-CNNs pre-trained on the local patches to take into account both local and holistic information. A deep mixture model is also proposed to further address the mis-alignment brought by pose variation. DMM captures the spatial-appearance distribution over faces to acquire the correspondences of the local patches. Without relying on the hand-crafted features, the proposed framework automatically learns an effective representation of face images to build an end-to-end system. We achieve the state-of-the-art performance with automatic feature learning in the two benchmark datasets in the wild. The proposed part-based framework is composed of two separate parts which are optimized with regard to two different objectives. It would be more intriguing if the two networks can be combined in a unified framework such that joint optimization is possible. More research works are expected on this perspective in the future.

Chapter 4

Generic Cross-Modality Face Recognition

The DMM-CFN framework in Chapter 3 addresses face verification in the wild with specific focus on local variations in terms of pose and illumination. In this chapter, the multi-modal face recognition problem is studied in a more general way. The proposed conditional Convolutional Neural Network (c-CNN) explores the hidden modalities of data directly, and is applicable for both face identification and verification problems. The basic assumption is that data may appear in different views or styles in computer vision. For example, objects of the same class may have different types in object recognition, e.g., cars may be of various types and brands; or in human pose estimation, people with the same pose may have different identities. Similarly, many face related tasks deal with images with variations in terms of pose, occlusion and lighting, and thus are inherently multi-modal. Such multi-modality issue leads to a large intra-class variation, which poses a great challenge to most existing approaches for face identification or verification.

A general approach to handle multi-modal problems is to find a shared feature space where data of different modalities are directly comparable. Conventional methods, such as Canonical Correlation Analysis (CCA) [Hardoon et al., 2004] and Partial Least Squares (PLS) [Geladi and Kowalski, 1986], aim at learning modality-specific projection matrices that lead to maximal covariance among instances of the same class in the shared latent space. Many works are specifically designed to deal with two-view data. In particular, data of one view are carefully projected into the subspace of the other modality. This idea has witnessed popular applications in synthesis based approaches for various problems, such as sketch-photo verification [Wang and Tang, 2009], low resolution vs. high resolution face matching [Liu et al., 2007], etc. Despite excellent work has been done on synthesis, this may in principle be an ill-posed problem that is more difficult than discriminatively comparing images of two different modalities.

Most of the aforementioned approaches are built on hand-crafted features. However, it is difficult to manually design features insensitive to the variations across modalities, since instances of different modalities usually span different feature spaces. In addition, the generic features, such as SIFT [Lowe and G, 1999], HOG [Dalal and Triggs, 2005] and LBP [Ojala et al., 2002a], are designed to solve certain problems, and thus may not be optimal for specific variations in the given problems. Moreover, some characteristic visual information may be lost in the extraction (especially the quantization) stage, which usually cannot be recovered in later stages. Recent deep learning methods [Sun et al., 2014b, Girshick et al., 2014, Lin et al., 2013], on the other hand, are able to learn an effective representation from raw-pixel inputs by directly optimizing with regard to the given objective. Deep learning also witnessed several attempts in handling cross-modality variations [Zhu et al., 2013, Zhu et al., 2014, Kan et al., 2014]. In most aforementioned approaches, the training or even testing instances come along with pre-defined modality information. For example, many approaches for multi-pose face recognition assume that the head pose is known during training. However, the ground-

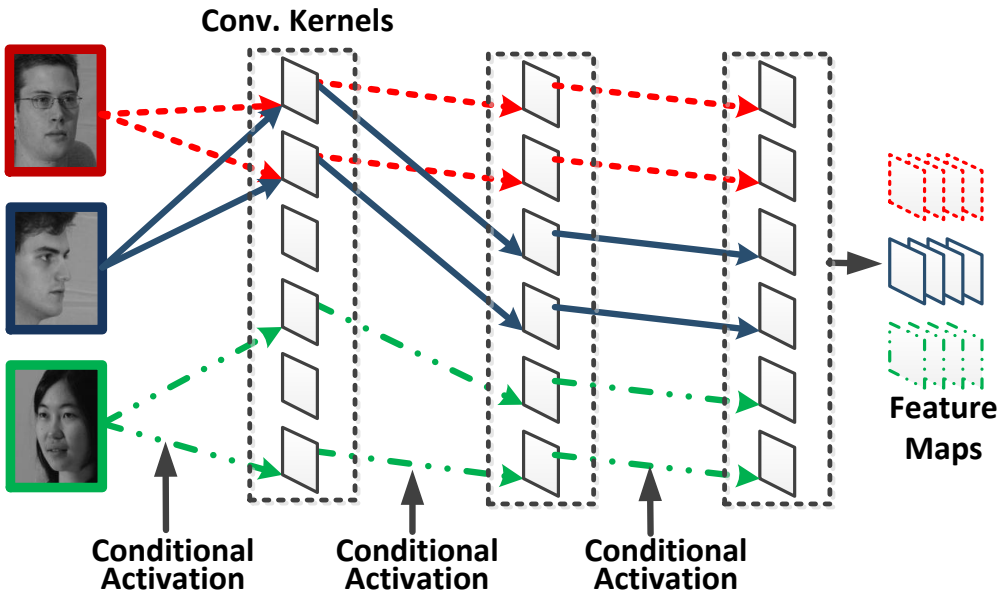

Figure 4.1: Illustration of c-CNN. Each line type stands for one modality. Each image is passed along with a modality-specific route indicated by the corresponding colored arrows. Only the kernels along the route are activated and utilized to extract features. The passing route defines the splitting with regard to inherent modalities in a coarse-to-fine manner: similar modalities, e.g., modality of red dashed line and blue solid line, may share certain kernels at the beginning layers.

truth modality information is not usually available in practice. Moreover, it is also possible that the modalities of data are vague and difficult to define explicitly when, e.g., faces appear with multiple variations in poses, illumination, expression, occlusion, etc.

In this chapter, we introduce a generic deep learning framework, termed as conditional Convolutional Neural Network (c-CNN), to address multi-modal classification problems with no prior knowledge on data modality. The proposed network automatically learns the inherent modality distribution and the feature representation with regard to an unified objective. In standard CNN, the convolution kernels for each sample are immutable during training, and all the input samples are processed with the same

kernels if no modality information is provided. In contrast, we include conditional computation of the "routes" for samples to propagate through the network. In particular, for each sample in one training epoch, the convolution kernels are sparsely activated within each layer, and the activated kernels across layers define a "route" for the given sample as shown in Figure 4.1. The activations of kernels in different layers are dependent and jointly optimized in a learnt manner. To be more specific, the activation probability of a kernel for a certain sample is conditioned on the corresponding intermediate representation and the routing status in the lower layers.

Conditional routing brings benefits in two folds: 1) the large intra-class variations across modalities make it very difficult to model the complex problem with an unified representation. The conditional routing gradually projects data of different modalities into several subspaces where the intra-class variations are much easier to be handled; 2) conditional routing activates only a limited number of convolution kernels in a learnt and optimized way. As a result, the computation cost is largely reduced, which makes the network more scalable. Decision tree inherently embeds the concept of conditional computation via hierarchical partitions, and thus is incorporated into CNN to substantiate the proposed framework. In particular, each tree node learns the intermediate representation and finds an optimal way to split samples at the same time. The proposed method is evaluated in two recognition problems of multi-modal faces, and proved to be effective with various comparisons.

4.1 Related Work

Multi-modality spans a wide range of research, and has been explored in a large number of prior works. Common approaches handle the variations across modalities via mapping samples into a shared latent space. Kim and Josef [Kim and Kittler, 2005] introduced a set of locally linear transformations to address multi-view face recognition.

The proposed method maximizes the separability of classes locally while promoting consistency between the multiple local representations of single class objects. Abhishek et al. [Sharma and Jacobs, 2011] used Partial Least Squares (PLS) to linearly map images of different modalities to a common linear subspace in which they are highly correlated. The proposed method is evaluated in cross-view, cross-resolution and sketch vs. photos face matching problems, and demonstrates considerable improvements over conventional methods. Abhishek et al. [Sharma et al., 2012] proposed the Discriminant Multiple Coupled Latent Subspace framework to handle cross-view face recognition. It learns a set of pose-specific projection directions such that the projected images of the same subject are maximally correlated in the target latent space. Kan et al. [Kan et al., 2012] followed a similar approach to handle multi-view object recognition. They jointly learn multiple view-specific linear transformation in a non-pairwise manner. In these papers, the global non-linear data structures are assumed to be linearly separable in the transformed local spaces. Motivated by the recent success of deep features [Krizhevsky et al., 2012,Lin et al., 2013], we propose to learn the required non-linear mappings within the latent local spaces with Deep Neural Network (DNN).

Many previous studies have also explored the approaches to synthesize faces of a certain modality in a statistic manner. Liu et al. [Liu et al., 2007] synthesized high-resolution face images from low-resolution images via integrating a global parametric model and a local non-parametric model. Wang and Tang [Wang and Tang, 2009] proposed a face photo retrieval system, which transforms a face image into a sketch. The proposed system conducts transformation on shape and texture of face images respectively. Zhang et al. [Zhang et al., 2006] targeted at face recognition with variations of illumination and pose. They proposed a texture synthesis method by employing a generic 3D face shape. Similarly, Li et al. [Li et al., 2012] transformed faces of multiple poses to their frontal view via 3D face registration. However, the cross-modality transformation is complex and difficult to learn since it usually requires the corresponding

samples in the target modalities to be available for each image, which is not always the case in practice. Therefore, the cross-modality synthesis could be an harder problem than the direct discriminative matching of multi-modal subjects.

Recent research on deep learning [Sun et al., 2014b, Girshick et al., 2014, Lin et al., 2013] stimulates many applications of deep models in recognition problems with multi-modality. Zhu et al. [Zhu et al., 2013] transformed faces under any pose and illumination to their canonical view. The proposed network learns the feature extraction layers and the reconstruction layer jointly. Kan et al. [Kan et al., 2014] also addressed the cross-pose problem with a reconstruction-based deep model. The model transforms faces of large view gradually to its frontal view layer by layer. Zhu et al. [Zhu et al., 2014] proposed a multi-task learning method to optimize the pose estimation and recognition objective in a joint manner. The results indicate that the pose information provides important clues in matching faces across views. In most aforementioned works (with either manual features or deep learning), the feature extraction or subspace transformation are defined or learnt specifically for each modality. Under such a framework, the modalities of data have to be pre-defined explicitly, or in other words, the modality which the data instance belongs to has to be known. In contrast, our framework defines a generic method in handling multi-modality problems without any prior knowledge on modality. Instead, the modality is learnt together with the feature representation in a deep model with conditional computation.

The cascade of sample splitting in decision tree embeds the idea of conditional computation, and is well explored by many tree-structured classifiers [Tu, 2005, Wu and Nevatia, 2007, Zhao et al., 2013, Jordan and Jacobs, 1994]. The fusion of decision tree (forest) and feature learning is also mentioned in a few recent works. Bulo and Kontschieder [Bulo and Kontschieder, 2014] aimed at finding the optimal split function at each node of the tree with MLP. However, the optimal splitting of samples is learnt in the traditional layer-by-layer manner. In other words, the optimization of the split network in a

node is isolated from the learning of both its parent node and the existing nodes in other branches. Fanello et al. [Fanello et al., 2014] attempted to learn the optimal filtering kernels and apply them to each data point. However, the filters are adopted as the PCA components learnt from noisy patches of multi-scale. The optimal filters are actually "chosen" from a random pool to minimize the energy functions of the nodes. Similar to [Bulo and Kontschieder, 2014], the split function is learnt separately for each node. Moreover, there is no joint learning of features and splitting nodes in either approach. In contrast, we jointly optimize the splitting nodes of the tree and the convolution kernels of the neural network with regard to an unified objective function. After our publication [Xiong et al., 2015b] corresponding to this chapter, Kontschieder et al. [Kontschieder et al., 2015] proposed a deep network which integrates decision forest as the final decision layer. Their framework proposed a stochastic and differentiable decision forest such that the layer-by-layer back propagation can be conducted. Different from their method, the leaf node and neural layers are tightly coupled in each layer of the proposed method. Moreover, our motivation is to use the conditional routing to explore the hidden modality so as to decompose the original problem into simple sub-problems. The conditional routing in this chapter should not be constrained to decision tree or forest.

4.2 Conditional Convolutional Neural Network

In this chapter, we assume that the given problem is potentially multi-modal, and the modality information is not known for either training or testing. This is a more general assumption in practice.

The inherent modality is explored via finding the optimal set of convolution kernels to be activated. For a given sample, only the corresponding activated convolution kernels are utilized to extract features. The activated kernels within each layer define a passing route for a given sample. Intuitively, training samples of the same modality

should follow the same route through the network. Traditional CNN activates all the kernels for all the training samples. For c-CNN, the activation of kernels in the layer i is jointly determined by the present input representation $\mathbf{X}_n^{(i)}$ to the layer i and the passing route in the preceding layers $\{\boldsymbol{\theta}_n^{(i)}, j = 0, ..., i-1\}$. In our implementation of tree-structured CNN, $\boldsymbol{\theta}_n^{(k)} \in \{0,1\}^k$, where 0 indicates the sample goes to the left node and 1 stands for the route to the right node.

We donote n as the index of input samples, and the corresponding forward function can be formulated as follows

$$\mathbf{X}_{n,k}^{(i+1)} = g_{n,k}^{(i)} \cdot \sigma(\widetilde{\mathbf{W}}_k^{(i)} * \mathbf{X}_n^{(i)} + b^{(i)}), \tag{4.1}$$

where $\mathbf{X}_{n,k}^{(i+1)}$ is the k-th kernel map of the n-th sample in layer $i+1$, and $g_{n,k}^{(i)}$ denotes the activation indicator of the k-th convolution kernel $\widetilde{\mathbf{W}}_k^{(i)}$. $g_{n,k}^{(i)}$ follows a Bernoulli distribution, i.e., $g_{n,k}^{(i)} \sim B\{1, p_{n,k}^{(i)}\}$, where

$$p_{n,k}^{(i)} = P(\theta_{n,k}^{(i)} | \mathbf{X}_n^{(i)}, \boldsymbol{\theta}_n^{(i-1)}, ..., \boldsymbol{\theta}_n^{(0)}), \tag{4.2}$$

and $\theta_{n,k}^{(i)}$ is the k-th element in $\boldsymbol{\theta}_n^{(i)}$. It should be noted that c-CNN aims at exploring the underlying modality distribution of data and the corresponding feature representation for each modality in an unified framework. In particular, the feature extraction parameters $\widetilde{\mathbf{W}}_k^{(i)}$ and $b^{(i)}$ and the kernel activation parameter $\boldsymbol{\theta}_n^{(i)}$ are learnt with regard to an unified objective function in a joint manner.

The conditional activation of convolution kernels can be defined in various ways. Decision tree embeds the concept of conditional computation in the hierarchy of simple decisions across layers and has seen plentiful applications in various fields. The leaf nodes in one layer are mutually exclusive, and each sample can be passed only to one leaf node. The choice of leaf nodes for certain input is conditioned on the split function of its parent nodes and the existing route in the above layers. The aforementioned characteristics make the decision tree a good option to realize c-CNN. In this chapter, the

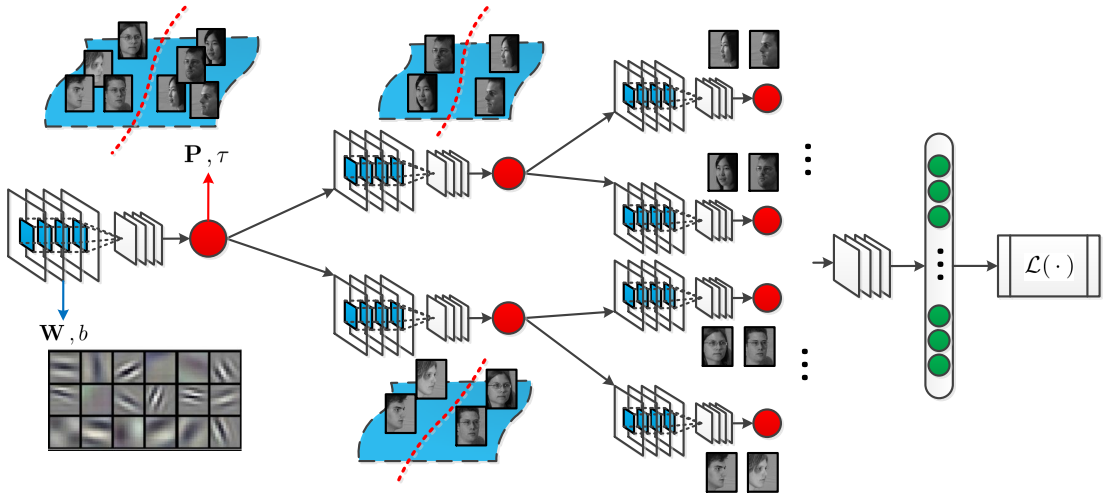

Figure 4.2: A specific example of c-CNN with Modality-aware Projection Tree (MPT). Each tree node computes the intermediate representation with CNN and the partition of samples in the projected latent space. With the help of MPT, samples of different modalities are gradually separated layer by layer and finally passed into the different leaf nodes. Both the features and the split functions are jointly optimized w.r.t. one unified loss function $\mathcal{L}$.

conditional computation of decision tree is incorporated into CNN as a specific instance of c-CNN. In particular, the tree nodes split the convolution kernels in each layer into several mutually exclusive kernel sets. However, there is no such a hard segmentation constraint for generic c-CNN. The assignments of convolution kernels are more flexible for a given input sample in generic c-CNN. Therefore, this decision tree based approach can be regarded as a simplified case. The proposed network includes two components – Modality-aware Projection Tree (MPT) and Convolutional Neural Branch (CNB). Detailed explanations are included for both components in the following subsections.

4.2.1 Modality-aware Projection Tree

Modality-aware Projection Tree (MPT) aims at defining a hard partition in the sample space such that samples of the same modality fall into the same leaf node. The modality is explored via learning of the split function for each node of the tree. To be more specific, we intend to learn the splitting of samples in an unsupervised manner such that the sample space is segmented with regard to the inherent modalities as illustrated in Figure 4.2.

Let's denote $\mathcal{X}$ and $\mathcal{Y}$ as the input and output space for a given classification problem. To begin with, we define a fully-grown decision tree of depth D. The node of the tree is denoted as $V^{(i,j)}$, where i is the index of the layer in the tree and j is the index of the leaf node in the i-th layer. Correspondingly, $\mathbf{x}_n^{(i,j)}$ is the intermediate representation of the sample $\mathbf{x}_n \in \mathcal{X}$.

Within the node $V^{(i,j)}$, the passing route of a sample is determined by a split function $h : \mathcal{S} \rightarrow \{\mathcal{S}^{\mathcal{L}}, \mathcal{S}^{\mathcal{R}}\}$, if we denote the whole input set for this node as $\mathcal{S}$, and the subsets of two child nodes as $\mathcal{S}^{\mathcal{L}}$ and $\mathcal{S}^{\mathcal{R}}$ respectively. The split function can be formulated as

$$\mathbf{x} \in \begin{cases} \mathcal{S}^{\mathcal{L}}, & \varphi(\mathbf{x}) \geq 0 \\ \mathcal{S}^{\mathcal{R}}, & \varphi(\mathbf{x}) < 0 \end{cases}. \tag{4.3}$$

In this section, MPT is constructed in a similar way as Random Projection Tree [Dasgupta and Freund, 2008]. The feature test function φ is defined with a projection vector $\mathbf{P}^{(i,j)}$ and a bias $\tau^{(i,j)}$ as follows,

$$\varphi(\mathbf{x}) = \mathbf{x}^T \cdot \mathbf{P}^{(i,j)} + \tau^{(i,j)}. \tag{4.4}$$

An unsupervised constraint is imposed for each node such that the distance between the centroids of two sub-clusters is maximized. The corresponding node-wise loss is

formulated as

$$\mathcal{J}_t = \frac{\frac{1}{N}\sum_{\mathbf{x}\in\mathcal{S}}\varphi(\mathbf{x})^2}{(\frac{1}{N_\mathcal{L}}\sum_{\mathbf{x}\in\mathcal{S}^\mathcal{L}}\varphi(\mathbf{x}) - \frac{1}{N_\mathcal{R}}\sum_{\mathbf{x}\in\mathcal{S}^\mathcal{R}}\varphi(\mathbf{x}))^2}, \tag{4.5}$$

where $N_\mathcal{L}$ and $N_\mathcal{R}$ are the numbers of samples falling into the left and the right child node respectively, and $N = N_\mathcal{L} + N_\mathcal{R}$.

4.2.2 Convolutional Neural Branch

Apart from the splitting of the input samples, each tree node also learns an intermediate representation with regard to the given objective directly. In particular, a tree node $V^{(i,j)}$ contains a standard convolutional layer $C^{(i,j)}$ with max-pooling.

When a sample is given at the root node of the tree, it is passed forward along a specific path. Along that path, the given sample is processed through a complete Convolutional Neural Network, named as Convolutional Neural Branch (CNB), at the same time. MPT is prone to constrain samples with the same inherent modality to follow the same path such that each CNB learns a modality-specific mapping to the shared latent feature space. Different from conventional approaches for learning modality-specific mapping, CNBs of different modalities can share certain intermediate nodes as in Figure 4.2. Our motivation is that samples of similar modalities should be processed more similarly than those of dissimilar modalities.

We denote $\mathbf{W}^{(i,j)}$ and $b^{(i,j)}$ to be the weight and bias of the convolutional layer for the node $V^{(i,j)}$. The corresponding forward function is defined as

$$\widetilde{\mathbf{X}}_n^{(i,j)} = \sigma(\mathbf{W}^{(i,j)} * \mathbf{X}_n^{(i,j)} + b^{(i,j)}), \tag{4.6}$$

where $*$ represents the convolution operator.

The hierarchical splitting of decision tree inherently takes into account the routing status in the previous layers. Accordingly, the conditional forward function in Eqn.(4.1)

is transformed as follows,

$$\begin{cases} \mathbf{X}_n^{(i+1,2j)} & = \mathbb{1}(\varphi(\widetilde{\mathbf{X}}_n^{(i,j)}) \geq 0) \cdot \widetilde{\mathbf{X}}_n^{(i,j)} \\ \mathbf{X}_n^{(i+1,2j+1)} & = \mathbb{1}(\varphi(\widetilde{\mathbf{X}}_n^{(i,j)}) < 0) \cdot \widetilde{\mathbf{X}}_n^{(i,j)} \end{cases}, \tag{4.7}$$

where $\mathbf{X}_n^{(i+1,2j)}$ and $\mathbf{X}_n^{(i+1,2j+1)}$ are the input representations for the two child nodes of $V^{(i,j)}$ respectively, and $\mathbb{1}(\cdot)$ represents an indicator function.

Network Configuration.

Throughout the whole chapter, we adopt the same network structure as shown in Figure 4.2. The depth of the decision tree is set as 3. Correspondingly, each CNB is a three-layered neural network – 20 convolution kernels in the first layer, 20 in the second and 40 in the third. The kernel size is set as 5×5 for the 1st and 2nd layer, and 3×3 for the last layer, respectively. The non-linearity function $\sigma(\cdot)$ in Eqn.(4.6) is defined as ReLu for all the convolution layers. Each convolutional layer is followed by a max-pooling operator with pooling size 2×2 and pooling stride 2×2. To regulate over-fitting, we adopt momentum, ℓ-2 norm regularization and dropout in the learning process. The momentum is set as 0.5, and linearly increased to 0.9 within 50 iterations. Dropout is adopted at each layer, and the dropout rate is 0.5 for multi-PIE and 0.2 for Occluded LFW respectively. We adopt a smaller drop rate for Occluded LFW since the number of training samples is much larger than that of multi-PIE, and the network suffers less from over-fitting. All the parameters (including those for the tree partitioning) are initialized by uniform sampling within the range [-0.1, 0.1]. The output feature maps of each neural branch are forwarded to a shared fully-connected layer L with 50 hidden units. The output of this layer is the final representation of input faces. An n-class softmax layer is then appended on the top for the given classification problem.

Computation Analysis.

As the depth of our decision tree is fixed as 3, we have 4 leaf nodes in the final layer. Compared with the conventional CNN of the same structure as one CNB in Figure 4.2, the proposed network appears to contain more parameters. However, each input sample is only passed through one possible CNB in each iteration. To be more specific, in each iteration the route for each sample is firstly computed based on the current state of the network, i.e., each sample is passed down to one leaf node first. Afterward, the parameters of each CNB are updated according to the samples that following the route in that CNB. Both the partition weights and the feature learning weights are updated in one iteration, thus the passing route of each sample can be different from the current route in the next iteration. As the route changes for samples are frequent in the early iterations, the loss shows an obvious turbulence. The turbulence finally disappears as the network gradually converges. Namely, the computation complexity for each input sample is the same as the conventional single-model CNN. For fair comparisons, we increase the width of the single-model CNN so that it can have the same number of parameters as ours – the baseline CNN has 20 filters in the 1st layer, 40 in the 2nd and 160 in the 3rd. The runtime complexity of the i-th layer in c-CNN is $\frac{N^{(i-1)}}{2^{i-2}} \cdot \frac{N^{(i)}}{2^{i-1}} \cdot O(conv.)$, and the complexity of the CNN baseline is $N^{(i-1)} \cdot N^{(i)} \cdot O(conv.)$, where $O(conv.)$ is the complexity of the convolution operation of one kernel over one feature map, and $N^{(i)}$ is the number of kernels in layer i.

4.2.3 Joint Learning of MPT and CNB

Different from prior works that learn features node-by-node in a decision tree [Fanello et al., 2014, Bulo and Kontschieder, 2014], the feature representation and the split func-

tion of all nodes are jointly learnt with regard to an unified objective as

$$\mathcal{J} = \sum_n \mathcal{J}_\ell(\mathbf{x}_n, y_n) + \beta \sum_i \sum_j \mathcal{J}_t^{(i,j)}, \tag{4.8}$$

where the first term represents the softmax loss for the n-class classification problem, and the second term is the node-wise loss defined in Eqn.(4.5), and β is a scaling factor.

The network is optimized via back propagation with Stochastic Gradient Descent method. For a node $V^{(i,j)}$, the network needs to update 4 parameters – $\mathbf{P}^{(i,j)}$, $\tau^{(i,j)}$, $\mathbf{W}^{(i,j)}$ and $b^{(i,j)}$. The gradient w.r.t. each parameter is given in details in the following. It is noted that the optimization is conducted in a batch-wise manner. In particular, we use the partition parameters $\mathbf{P}^{(i,j)}$ and $\tau^{(i,j)}$ learnt with the previous data batch to split samples in the present batch. In this way, the dynamic routing of samples is determined before updating the parameters in the present batch iteration. To compute the gradient w.r.t. $\mathbf{W}^{(i,j)}$ and $b^{(i,j)}$, we need to derive the gradient w.r.t. $\widetilde{\mathbf{X}}_n^{(i,j)}$ first,

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \widetilde{\mathbf{X}}_n^{(i,j)}} = \frac{\partial \mathcal{J}_\ell}{\partial \widetilde{\mathbf{X}}_n^{(i+1,2j)}} \cdot \mathbb{1}(\varphi(\widetilde{\mathbf{X}}_n^{(i,j)}) \geq 0) + \\ \frac{\partial \mathcal{J}_\ell}{\partial \widetilde{\mathbf{X}}_n^{(i+1,2j+1)}} \cdot \mathbb{1}(\varphi(\widetilde{\mathbf{X}}_n^{(i,j)}) < 0) + \beta \frac{\partial \mathcal{J}_t^{(i,j)}}{\partial \widetilde{\mathbf{X}}_n^{(i,j)}}.\end{aligned} \tag{4.9}$$

Based on Eqn. (4.9), $\frac{\partial \mathcal{J}}{\partial \mathbf{X}_n^{(i,j)}}$, $\frac{\partial \mathcal{J}}{\partial \mathbf{W}_n^{(i,j)}}$ and $\frac{\partial \mathcal{J}}{\partial b_n^{(i,j)}}$ can be easily derived similarly as standard CNN with the chain rule. The splitting parameters can be updated as follows,

$$\frac{\partial \mathcal{J}}{\partial \mathbf{P}^{(i,j)}} = \frac{\partial \mathcal{J}_\ell}{\partial \mathbf{X}_n^{(i+1,\cdot)}} \cdot \frac{\partial \mathbf{X}_n^{(i+1,\cdot)}}{\partial \mathbf{P}^{(i,j)}} + \beta \frac{\partial \mathcal{J}_t^{(i,j)}}{\partial \mathbf{P}^{(i,j)}}, \tag{4.10}$$

$$\frac{\partial \mathcal{J}}{\partial \tau^{(i,j)}} = \frac{\partial \mathcal{J}_\ell}{\partial \mathbf{X}_n^{(i+1,\cdot)}} \cdot \frac{\partial \mathbf{X}_n^{(i+1,\cdot)}}{\partial \tau^{(i,j)}} + \beta \frac{\partial \mathcal{J}_t^{(i,j)}}{\partial \tau^{(i,j)}}. \tag{4.11}$$

Since $\frac{\partial \mathbf{X}_n^{(i+1,\cdot)}}{\partial \mathbf{P}^{(i,j)}}$ and $\frac{\partial \mathbf{X}_n^{(i+1,\cdot)}}{\partial \tau^{(i,j)}}$ are all zeros, the gradients are actually determined by the tree node loss $\mathcal{J}_t^{(i,j)}$, i.e.,

$$\frac{\partial \mathcal{J}}{\partial \mathbf{P}^{(i,j)}} = \beta \frac{\partial \mathcal{J}_t^{(i,j)}}{\partial \mathbf{P}^{(i,j)}}, \tag{4.12}$$

$$\frac{\partial \mathcal{J}}{\partial \tau^{(i,j)}} = \beta \frac{\partial \mathcal{J}_t^{(i,j)}}{\partial \tau^{(i,j)}}. \tag{4.13}$$

To simplify the problem, τ can be set as the mean value of samples after projections, i.e., $\varphi(\mathbf{x}) = \mathbf{x}^T \cdot \mathbf{P}^{(i,j)}$, such that there are only three parameters to optimize.

4.3 Relationships with Other Works

This section further discusses about the relationships between c-CNN and two following concepts briefly.

Connection with Ensemble Learning

Ensemble learning relies on the complementary effects of multiple weak classifiers. The fusion of such weak classifiers usually leads to a better generalization performance. Many popular methods, such as random forest [Breiman, 2001] and boosting [Freund and Schapire, 1999], fall into the broad category of ensemble learning. Within the structure of c-CNN, there exists multiple CNBs each of which corresponds to a certain kind of hidden modality. When passing through a CNB, a sample is examined against a simple partition test at each node, which can be viewed as a weak classifier. Moreover, each CNB defines a modality-specific mapping, and thus can also be viewed as a weak classifier. Similar to bagging, different CNB is learnt with different sub-set sampled from the original dataset. In such a sense, c-CNN can be regarded as a special variant of ensemble learning that integrates weak classifiers of two kinds.

Connection with Dropout

Dropout is a common strategy to suppressing over-fitting for DNN. In standard dropout, each dimension of the output feature maps is randomly set as zero accord-

ing to a pre-defined probability. In other words, the random activations are applied in terms of the feature dimension for dropout. In contrast, c-CNN is different from dropout on the following two aspects. To begin with, the activations are defined at the level of convolution kernels instead of feature dimension. Secondly, c-CNN learns the partition of data from the intermediate feature so as to determine the dropout of convolution kernels instead of random dropout.

4.4 Experiments

Our method is evaluated with two problems: 1) multiview face identification on Multi-PIE dataset [Gross et al., 2010] and 2) occluded face verification on Labeled Faces in the Wild dataset (LFW) [Huang et al., 2007b] with synthetic occlusions. The proposed c-CNN is built on a basic assumption that the modality information is unknown for both training and testing. Therefore, we do not include the comparison with some existing methods using the specific modality information of each sample. Experimental results are analyzed in details in the following subsections.

4.4.1 Experiment Settings

On both datasets, we use the same network configuration as shown in section 4.2.2. The implementation of c-CNN is based on Theano[1] and Pylearn2[2]. The supervised cost $\mathcal{J}_\ell(\cdot)$ in both experiments is the negative likelihood of an n-class softmax function, and thus n is set as 150 and 2 for multi-PIE and occluded LFW respectively. With more classes, the initial cost is much larger in scale. To balance the relative effect of the supervised cost and tree node cost, β is set to 5 and 1 accordingly. As most CNNs are optimized with batch-based SGD, the tree node loss in Eqn.(4.5) is only defined in batches.

[1] http://deeplearning.net/software/theano/
[2] http://deeplearning.net/software/pylearn2/

Thus, larger batch size can lead to better results. In this chapter, all the experiments are conducted with GTX TiTan GPU with 3GB memory. Due to the memory limit, we set the batch size as 1,000 in the following experiments.

4.4.2 Multi-View Face Identification

We evaluate the performance of c-CNN in multi-view face identification on Multi-PIE. It contains images of 337 identities with 20 illumination levels and 15 poses ranging from -90° to $+90^\circ$. The database is arranged in four sessions, and we evaluate our method on Session 1 only, which includes faces of 250 subjects. Previous experiments reported on Multi-PIE are usually conducted on faces with small poses (-45° to $+45^\circ$). However, our method is testified on faces under all poses. We follow a similar evaluation protocol as in [Zhu et al., 2013]. For training, we utilize all the images (15 poses, 20 illumination levels) of the first 150 identities. For testing, we choose one frontal image with neutral illumination marked as ID 07 as the gallery image for each of the remaining 100 subjects. The remaining images are used as probes. The average precision is reported for comparison with regard to pose in Table 4.1.

	Avg.	±90°	±75°	±60°	±45°	±30°	±15°	pose
Fisher Vector [Simonyan et al., 2013]	66.60	24.53	45.51	68.71	80.33	87.21	93.30	×
FIP_20 [Zhu et al., 2013]	67.87	34.13	47.32	61.64	78.89	89.23	95.88	√
FIP_40 [Zhu et al., 2013]	70.90	31.37	49.10	69.75	85.54	92.98	96.30	√
CNN_40	70.81	32.08	47.79	69.48	85.99	93.04	96.60	×
Cluster_CNN	69.87	36.80	47.36	68.20	82.43	90.67	93.75	×
Tree_CNN	71.16	39.90	50.29	67.21	83.63	91.31	94.66	×
c-CNN	73.54	41.71	55.64	70.49	85.09	92.66	95.64	×
c-CNN Forest	76.89	47.26	60.66	74.38	89.02	94.05	96.97	×

Table 4.1: Comparisons of precision (%) with some prior methods on multi-PIE for different poses. The last column indicates the dependency on head pose information.

Four methods are included for comparison in this subsection. Fisher Vector [Simonyan et al., 2013] is built on hand-crafted features, i.e., SIFT and LBP in this experiment. Both FIP [Zhu et al., 2013] and CNN_40 are deep learning based methods. We include the results of FIP with two network configurations. FIP_20 has exactly the same number of convolution kernels as one CNB in c-CNN. FIP_40 is included to show the improvements of c-CNN over the network with the same total number of parameters. In particular, FIP_40 has 20 kernels in layer 1, 40 in layer 2 and 160 in layer 3. FIP is a reconstruction based approach, and thus requires the frontal view and neural illumination for each image during training. In this experiment, we apply PCA on features of the last convolution layer such that the final dimension is the same as c-CNN. CNN_40 is a single CNN network with the same configuration of convolutional layers as FIP_40. Note that although our network has approximately the same number of parameters, the computation cost is much lower as analyzed in Section 4.2.2. Clearly, c-CNN achieves the best performance, especially for large poses, such as $\pm 90^\circ$ and $\pm 90^\circ$. The improvements can be up to nearly 10%. Different from FIP which requires frontal images for each subject, we do not utilize any pose information and our method still reaches higher accuracy. Moreover, c-CNN outperforms both CNN_40 and FIP_40 while maintaining a much lower computation cost. Moreover, we include two extra baselines – Cluster_CNN and Tree_CNN. Cluster_CNN firstly clusters the samples based on LBP features and trains a separate CNN for each cluster. Tree_CNN follows the c-CNN structure, but optimizes the branching parameters w.r.t. the node-wise loss first, and then learns the parameters of CNN while fixing the branching parameters. The improvement brought by c-CNN demonstrates the effectiveness of joint optimization over filters and tree branching.

In this subsection, we also explore the possibility of extending tree to forest as shown by c-CNN Forest. For this approach, we include 3 trees with $\beta = 5$, 7 and 10 respectively. In c-CNN Forest, we take the average of the cosine distance matrices of the derived corresponding feature vectors. As can be observed in the table, the performance is further

Figure 4.3: Partitioned samples of multi-PIE in leaf nodes. The blue boxes represent the tree nodes in the second layer, and the red ones stand for those in the third layer. The node notations are given inside the corresponding boxes. Clearly, samples of similar modalities (poses) are prone to be passed into the same nodes.

improved by more than 3%. Further randomization on parameters and bagging in the forest are expected to produce better results.

In addition, we illustrate some of the samples in each leaf node in Figure 4.3. Without any human intervention, the proposed method automatically discovers the inherent modality of the data (pose in this experiment) and clusters samples with similar poses into corresponding leaf nodes. Since the intermediate representation and splitting pro-

jections are jointly optimized w.r.t. Eqn.(4.5), the acquired clusters rarely contain noisy samples.

4.4.3 Face Verification with Various Occlusions

We evaluate c-CNN with occluded face verification on a synthesized dataset from Labeled Faces in the Wild dataset (LFW) [Huang et al., 2007b] – occluded LFW. LFW is a standard database collected to evaluate benchmark algorithms for face verification. It contains 13,000 images of 5,749 individuals downloaded from the Internet. We follow the image-restricted protocol of LFW. All the algorithms are evaluated with 6,000 pre-defined image pairs. The data are divided into 10 mutually excluded folds. In each experiment, data of only one fold are used for testing, and the remaining 9 folds are used for training.

In occluded LFW, each face image of LFW is synthesized with 6 kinds of occlusions, including hair, hand, mask, mustache, painting and glass. Each category includes 16 images occluded by the corresponding object. We crop the occlusion objects from a large collection of images from the Internet. Afterward the occlusions of objects are appended on the face images with reference to the detected landmarks. Some examples of the occluded faces are illustrated in Figure 4.4. Due to the large size of the dataset, we use a subset to evaluate the proposed network. In particular, for each image within a pair in standard protocol, we randomly sample 8 occluded images. The resulting two groups of images are then randomly combined to form 8 occluded pairs. This procedure is conducted for each fold.

Five baselines are included for comparison in this set of experiments. The results are reported on each fold in terms of the average precision in Table 4.2. HDLBP [Chen et al., 2013], Fisher Vector [Simonyan et al., 2013] and PEM [Li et al., 2013] are implemented with hand-crafted features. The aforementioned methods follow the same training pro-

Figure 4.4: Examples in Occluded LFW. Six categories of occlusions are synthesized for each image, including hair, hand, mask, mustache, painting and glass.

tocols (with no outside data) for fair comparison. We also include the single CNN based methods with the same network structure as one neural branch, i.e., CNN_20. As shown in the table, c-CNN demonstrates consistent improvements over CNN_20 and CNN_40, up to 3.5%. The significant improvements over CNN_20 can better demonstrate the superiority of the proposed method, since the two methods are of comparable computation cost. The improvements brought by c-CNN are further analyzed by showing some of the examples of corrected image pairs in Figure 4.6. Compared with modality-unaware CNN, c-CNN is more capable of modeling the intra-class similarities across different modalities. The synthesized data are very challenging due to the large occlu-

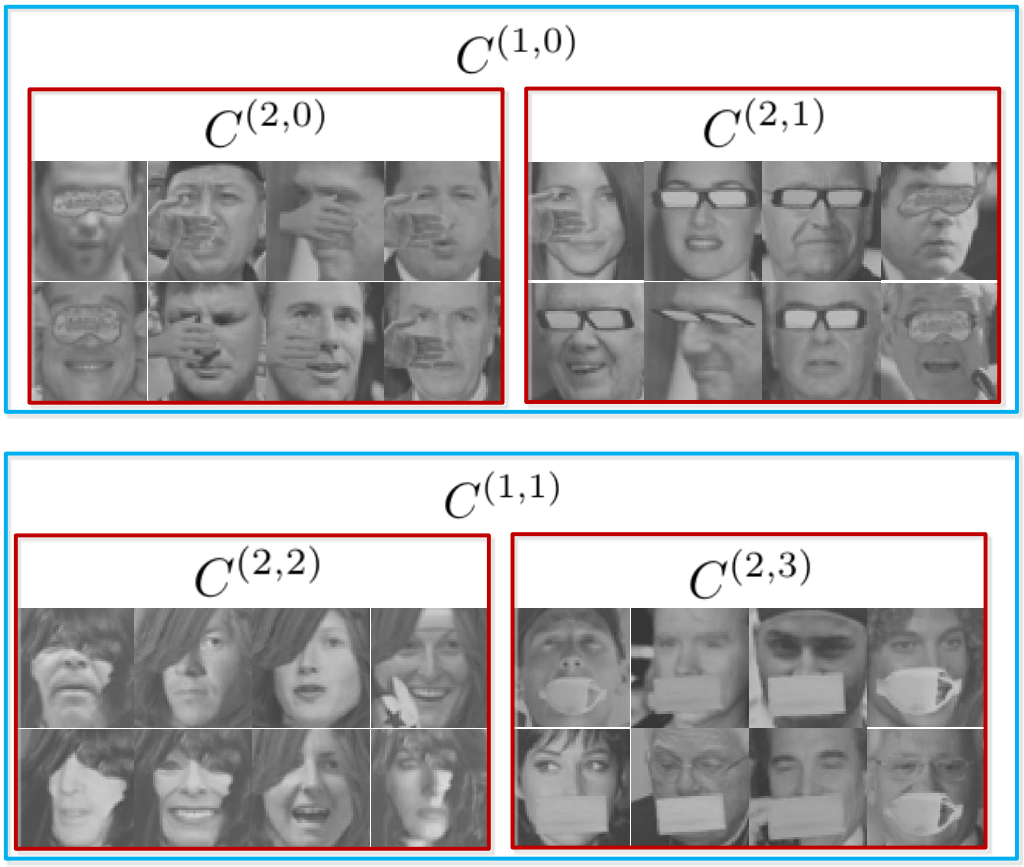

Figure 4.5: Partitioned samples of occluded LFW in leaf nodes. The blue boxes represent the tree nodes in the second layer, and the red ones stand for those in the third layer. Clearly, samples of similar modalities (occlusion categories and positions) are prone to be passed into the same nodes.

sion area on the faces, thus most manually designed features result in low precision. By including deep feature learning, c-CNN outperforms HDLBP, Fisher Vector and PEM on all the folds. As for the extension to the forest structure, we include 3 trees with β = 0.7, 1.0 and 1.2 respectively. The final score for each sample is computed as the maximum among the scores of each tree. The resulting performance is further improved by around 0.7%.

	1	2	3	4	5	6	7	8	9	10	Avg.
HDLBP [Chen et al., 2013]	69.77	68.79	66.39	69.09	67.45	66.89	67.70	67.26	66.71	69.85	67.99
Fisher Vector [Simonyan et al., 2013]	70.83	72.90	73.21	72.83	71.80	73.44	73.33	72.29	72.96	73.29	72.68
PEM [Li et al., 2013]	62.87	65.08	65.44	63.17	62.70	65.50	63.08	61.58	64.46	63.81	63.76
CNN_20	74.40	73.12	71.69	72.94	71.38	74.65	72.63	74.63	71.27	72.40	72.91
CNN_40	75.40	73.83	74.12	73.30	72.74	76.20	72.36	76.20	71.43	73.50	73.90
c-CNN	77.63	75.09	75.00	75.03	73.69	76.55	76.16	76.85	74.80	74.43	75.52
c-CNN Forest	77.65	75.16	75.00	76.17	73.71	77.67	77.27	77.81	76.10	75.83	76.24

Table 4.2: Comparisons of precision (%) with some prior methods on occluded LFW for ten folds.

Figure 4.6: Exemplars of the corrected image pairs by c-CNN.

Some of the examples in each leaf node are illustrated in Figure 4.5. With the exactly same setting as in multi-view face identification experiment, c-CNN discovers the inherent modality of input samples accordingly. It shows that the modality information is learnt as the occlusion type and position in this experiment. We also illustrate in Figure 4.6 some pairs corrected by applying the modality-specific partition of c-CNN. As observed from the figure, the improvement of accuracy is mainly caused by separating samples of distinct modalities, which is consistent with our initial motivation.

4.5 Conclusions

This chapter introduces a conditional Convolutional Neural Network to address cross-modality face recognition. By introducing conditional routing, c-CNN simultaneously explores the hidden modalities of samples and learns the modality-specific features

while maintaining a low computation cost. Both the conditional routing and the feature extraction are learnt optimally with the direct guidance of an unified loss. We evaluate c-CNN with decision tree in two cross-modality classification problems. In both experiments, c-CNN demonstrates consistent improvements. As a generic framework in handling cross-modalities, c-CNN can be easily applied in various research fields and we are expecting similar results as those in this chapter. Moreover, the decision tree based approach is a simplified case of c-CNN, which divides the convolutional kernels into mutually exclusive sets. In the future, we shall pursue a more generic c-CNN that enables flexible (soft) assignments of convolution kernels in each layer.

Chapter 5

Semi-Supervised Learning with Video Context

In Chapter 3 and 4, the proposed networks set major focus on addressing the variations in terms of modalities with limited data in a specific scenario. From another point of view, it will be much easier to train a classifier robust to various variations if we are given enough labeled data covering as many modalities as possible. Accordingly, this chapter tackles the problem of multi-modal face recognition from the perspective of scarcity of labeled samples. This idea is realized and evaluated in the problem of automatic character identification, which emerges with the explosive development of social network and video sharing websites. The purpose of automatic character identification is to associate character faces in photo albums or movies with names. Among many applications of character identification, celebrity-related tasks draw the most attention due to the public interest in celebrities. The faces of celebrities on the Internet are usually captured in the natural environment, thus are suitable subjects for the study of unconstrained face

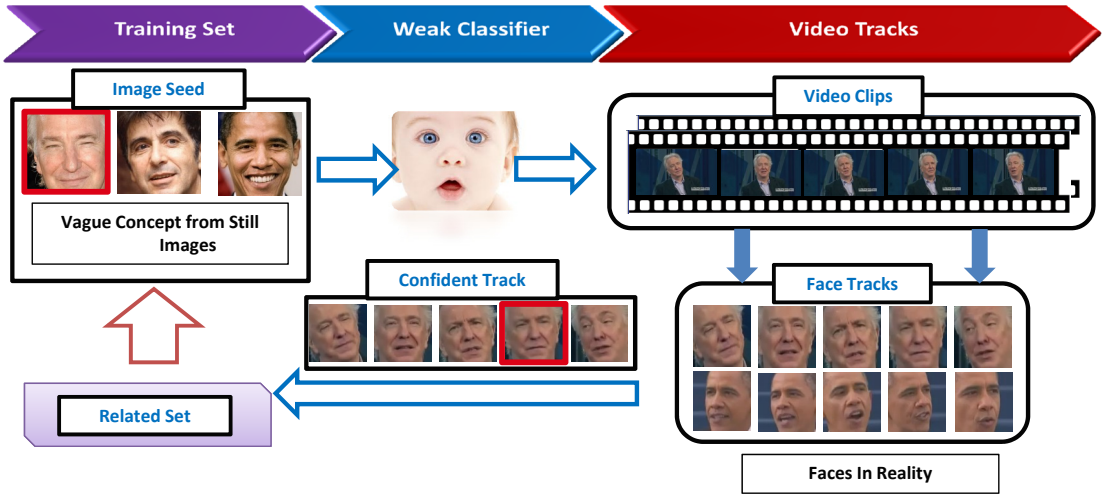

Figure 5.1: Illustration of the proposed adaptive learning framework. The initial classifier is trained on a small set of static images (image seeds), and then used to label the frames within each video track. If a certain frame is assigned with a confident label, all the frames within the same track are promoted into the *related set* and utilized to update the classifier in the next iteration such that the classifier gradually evolves.

recognition problem. Furthermore, celebrity identification has been considered as a crucial step for image/video semantic analysis [Ballan et al., 2010, Bertini et al., 2006, Satoh et al., 1999] in multi-media technologies.

Up to now, researchers have proposed many methods for celebrity identification [Berg et al., 2004, Zhao et al., 2008, Zhang et al., 2012]. Nevertheless, as mentioned in [Arandjelovic and Zisserman, 2005] the problem still remains tremendously challenging due to: 1) lack of precisely labelled training data; 2) significant visual variations in terms of human pose, light, facial expression, etc.; 3) low resolution, occlusion, nonrigid deformation, large motion blur and complex background in the realistic photographic conditions.

An intuitive way to deal with these challenges is to collect a large-scale face database with sufficient data diversity and reliable ground-truth labels. However, the enormous amount of manual work required in data labeling hinders constructing such a dataset.

On the other hand, the rapid development of the Internet provides easy access to a large collection of unlabeled face data. Commercial search engines, such as Google, can return a large pool of images corresponding to a certain celebrity within just several milliseconds. YouTube, a popular video sharing platform, receives around 100 hours of videos uploaded every minute. The massive data with easy accessibility have motivated researchers to investigate how to improve the performance of traditional learning based multimedia analysis methods utilizing such large volume of unlabeled data. As a result, Semi-Supervised Learning (SSL) [Zhu et al., 2003b, Zhou et al., 2003, Belkin et al., 2006] has drawn plenty of research interest during the past a few decades.

In this chapter, we propose to utilize video context to improve the accuracy for celebrity identification with only limited labeled images. Compared with faces returned by search engines, faces in videos involve higher diversity of variation, and thus are more similar to the realities. Although noisy, videos are usually accompanied by certain context information that can be used for de-noising. Accordingly, we extract face tracks from the downloaded videos and build the celebrity identification framework with a simple but effective assumption, i.e., faces from the same face track belong to the same celebrity. In particular, the proposed system firstly learns a weak classifier from a few labeled static images. The classifier is then applied on each face track to predict the labels and confidence scores of all the frames. The frames are ranked with regard to the confidence scores and the track possessing the frame with highest confidence score is chosen as a *confident* track. The video constraint enables the propagation of predication labels across the frames of the confident track, which is then promoted into the *related set*, as illustrated in Figure 5.1. The update of the classifier is realized under the supervision of *related samples* in the *related set*. This select-update process is iterated for multiple times such that the classifier evolves with improved discriminative capacity gradually. The proposed learning theme has certain analogy to some recent biological studies of the cognitive process of human brains. According to the Adaptive Resonance Theory

(ART) [Grossberg, 2013], human brains form the resonant states depicting the links between visual inputs and semantics in the initial learning stage, and then search for the good enough matches to enhance the understanding of objects or people gradually.

The proposed method shares certain similarity to self-training [Yarowsky, 1995, McClosky et al., 2006] since both of them adopt a mechanism of iteratively selecting samples from the unlabeled set to improve the performance. The difference lies in that our approach introduces the video context constraint into the selection process, such that the positive samples that cannot be recognized confidently may still be promoted. Self-training often suffers from the well-known semantic drifting issue [Shrivastava et al., 2012]. It occurs when the size of the labeled set is too small to constrain the learning process. More specifically, the errors in selecting the *best* samples may accumulate, thus the selected examples tend to stray away from the original concept. Existing solutions to semantic drifting mainly focus on improving the accuracy in the selecting process. Typical approaches following this theme includes co-training, active learning, etc. This chapter, on the contrary, explores from a different perspective. Instead of struggling to select the correct samples, we aim to design a classifier robust to the selection errors by treating the selected samples as *related* rather than "labeled". For this purpose, we decrease the influence of the selected samples, termed as *related samples*, to guarantee that their influence is weaker than labeled samples. Furthermore, the influence of a specific *related sample* is re-weighted based on the corresponding confidence score, so that discriminative samples are emphasized while noisy and non-discriminative samples are suppressed at the same time.

5.1 Related Work

Celebrity identification is a specific application of face recognition. Previous works on celebrity identification can be generally categorized into two groups: a) face recogni-

tion considering correspondence between face and text information; b) face recognition utilizing a large manually labeled image or video training set.

In the first group, the textual information is used to provide extra constraint in the learning process. An early work of Satoh et al. [Satoh et al., 1999] introduced a system to associate names located in the sound track with faces. Berg et al. [Berg et al., 2004] built up a large dataset by crawling news images and corresponding captions from Yahoo! News. Everingham et al. [Everingham et al., 2006] explored textual information in scripts and subtitles and matched it with faces detected in TV episodes. However, the main disadvantage in the studies of the first group is the heavy dependence on associated textual information. In most cases, nevertheless, the assumption that textual information is available does not hold, and errors may occur in the given text description.

The other group aims at learning a discriminative model based on a manually labeled dataset. For example, Tapaswi et al. [Tapaswi et al., 2012] presented a probabilistic method for identifying characters in TV series or movies. They trained a face model and a speaker model on several TV episodes with manual labels. In the work of Liu et al. [Liu and Wang, 2007], a multi-cue approach combining facial features and speaker voice models was proposed for major cast detection. However, the performance of supervised learning methods mentioned above was usually constrained by the insufficiency of labeled training samples. Thus, many researchers are more interested in scenarios where only a limited number of labeled training samples are available, which are much more common in reality.

Correspondingly, Semi-Supervised Learning (SSL) based methods are proposed in many studies [Zhu et al., 2003b, Zhou et al., 2003, Belkin et al., 2006]. These methods usually assumed that unlabeled data contain the information of underlying distribution and thus can facilitate the learning process. The video sharing websites, such as

YouTube, provides easy access to such a large unconstrained and unlabeled training set. Many studies have been conducted using video data in multiple active fields of computer vision, including object detection [Yang et al., 2013, Prest et al., 2012], object classification [Yan et al., 2006], person identification [Bauml et al., 2013], action recognition [Chen and Grauman, 2013], and attribute learning [Choi et al., 2013].

Among various SSL methods, one of the classic is the bootstrapping based method, also known as self-training. For instance, Cherniavsky et al. [Cherniavsky et al., 2010] trained a classifier on a set of static images and then applied it to classify attributes in videos. Chen et al. [Chen and Grauman, 2013] addressed the action recognition task by learning generic body motion from unconstrained videos. In their example-based strategy, the most confident pose is located in a nearest-neighbor manner and then added into the training set. Kuettel et al. [Kuettel et al., 2012] proposed a segmentation framework on the ImageNet dataset by recursively exploiting images segmented so far to guide the segmentation of new images. Choi et al. [Choi et al., 2013] learnt from confident attributes from unlabeled samples to expand the visual coverage of training sets. It also claimed that even though some attributes were selected from relevant categories, they could lead to improvement for category recognition.

A typical issue of the self-training methods is caused by the error in labeling confident samples in each iteration. To be more specific, early errors will accumulate by including more and more false positive samples, causing semantic drifting as mentioned in [Shrivastava et al., 2012]. Most researchers attempt to increase the labeling accuracy in selection to address semantic drifting. Standard approaches include active learning [Fathi et al., 2011] and co-training [Blum and Mitchell, 1998]. Active learning iteratively queries the supervision of the users on the least certain samples. Li and Guo [Li and Guo, 2013] proposed an adaptive active learning method by introducing a combined uncertainty measurement. They selected the most uncertain samples to query users' supervision. These selected samples are added into the training set and used to

re-train the classifier. Co-training or multi-view learning, on the other hand, learns a classifier on several independent feature sets or views of data [Blum and Mitchell, 1998] or learns several different classifiers from the same dataset [Goldman and Zhou, 2000]. Saffari et al. [Saffari et al., 2010] proposed a multi-class multi-view learning algorithm, which utilized the posterior estimation of one view as a prior for classification in other views. In [Minh et al., 2013], Minh et al. introduced RKHS of vector-valued functions into manifold regularization and multi-view learning, and achieved the state-of-the-art performance.

Incremental learning or online learning [Grabner and Bischof, 2006, Grabner et al., 2008] also includes a mechanism of iteratively updating the classifier. A common assumption is that the training samples with labels are given in a streaming manner, i.e., not all the training samples are presented at the same time. Incremental learning cannot select the confident unlabeled data as in self-training and its performance is quite sensitive to the label noises. In this chapter, we focus on learning a robust classifier with noisy selected samples. Thus, incremental learning is out of scope in this chapter.

We propose an adaptive learning approach for celebrity identification by incorporating the video context information. Moreover, we introduce the concept of *related sample* to address the problem of semantic drifting. Instead of struggling to prevent the error in labeling the unknown samples, we aim to obtain a classifier that is robust to selection errors such that the performance can be improved steadily.

5.2 Overview of Adaptive Learning

Adaptive Resonance Theory (ART) [Grossberg, 2013] is a cognitive and neural theory to describe how the brain learns to categorize in an adaptive manner. According to ART, human brain initializes the resonant states, which links the visual inputs to semantics,

via "supervised learning" and then tries to find "good enough" matches for the concepts in everyday life. These matches are then used for updating the resonant states in the learning process.

According to ART, a baby may learn in a two-stage manner – initial learning and adaptive learning.

- Initial Learning. A new born baby has not much knowledge, i.e., resonant states, of recognizing a certain object or person. Parents, acting as supervisors, show the baby the links between words (labels) and visual information, i.e., provide some initial labeled samples.
- Adaptive Learning. The baby observes the world by himself/herself. When a certain status of an object/person matches with the initial pictures in the brain (good match), it connects all the visual information of this object/person with the existing knowledge to update.

Sharing a similar idea, our framework includes a two-stage learning mechanism on a training dataset consisting of: a) labeled images for initial learning and b) unlabeled noisy data from the Internet for adaptive learning. The images are retrieved from Google image using the name of each celebrity as the query word and then manually labeled. For collecting the noisy data, we download video clips from YouTube with tags relevant to each celebrity. Faces in the static images online are usually taken under similar conditions, e.g., similar pose, facial expression and illumination. However, faces in the videos present more variations and thus provide more diverse training samples for the phase of adaptive learning. Note that the collected videos are noisy due to 1) the videos may not be relevant to the celebrity and wrongly selected due to the tagging errors of the users and 2) each video may contain several individuals. Thus such videos are treated as unlabeled data and fed into the classifier without using the ground-truth identities during training.

In this chapter, we extract multiple face tracks from the collected videos and exploit the video context information within the face tracks. We introduce the video constraint into the adaptive learning process – faces from the same track belong to the same identity. The video constraint has a natural connection with the "baby learning" process, as mentioned in the above section. The visual perception of the baby is continuous and the baby is able to tell the correspondence between the consecutive frames, i.e., whether these frames share the same identity. Namely, the baby organizes the visual perceptions in the real world as tracks of consecutive frames that belong to the same identity. The proposed video constraint utilizes the similar concept in a sense.

Before introducing the details of our methods, some notations are defined firstly for formal description. Suppose we are given in total n training samples of N individuals, which include l labeled samples and u unlabeled samples arranged in video tracks, i.e., $n = l + u$. The initial labeled image set is denoted as $\mathcal{L}_o = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$, where y_i represents the label for the sample $\mathbf{x}_i$. The unlabeled video set consists of K face tracks $\{\mathcal{T}_i \,|\, i \in \{1, 2, \ldots, K\}\}$ with $K \leq u$, and is denoted as $\mathcal{U} = \{(\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u})\}$. Here $\{\mathbf{x}_i, i = l+1, \ldots, n\}$ are the extracted frames from face tracks.

The most straightforward way to use the unlabeled samples is to treat the most confident unlabeled track $\mathcal{T}_i$ as labeled based on the corresponding confidence score. Here the confidence score can be computed based on the classifier learned from a few labeled samples. These tracks are termed as confident tracks, which correspond to the "good enough matches" in baby learning process [Grossberg, 2013]. All the frames within are then assigned with the same label as the most confident frame and promoted into the *related set* denoted as $\mathcal{L}_r$ (more details in Section 5.3). Afterwards, the classifier is retrained with the current "labeled set", the union of initial labeled set and discovered related set, $\mathcal{L} = \mathcal{L}_o \cup \mathcal{L}_r$. The updated classifier then predicts the labels of all the remaining frames in the video set. To identify multiple celebrities, the classifier is trained in a one-vs-all manner. More specifically, we train N binary classifiers, each of which

is learned by taking one class of samples as positive and the remaining $N-1$ classes of samples as negative. The most confident tracks are then selected per class in each iteration. Compared with selection for only one class, the per-class selection is aimed to avoid the dominance of a certain class in the track selection and balance the response magnitude of all the classifiers. The confidence score of each frame belonging to class j is computed via a soft-max function $g_j(\cdot)$ on the response of each classifier:

$$g_j(\mathbf{x}_i) = \frac{\exp\{f_j(\mathbf{x}_i)/\eta\}}{\sum_k \exp\{f_k(\mathbf{x}_i)/\eta\}}, \tag{5.1}$$

where $f_j(\cdot)$ denotes the binary classifier for the class j and η is a trade-off parameter for approximating the max function. Large η renders almost the same scores for different inputs, while small η enlarges the gaps among the output confidence scores.

We compute the confidence scores of all the frames within each face track. The maximum of these confidence scores within each track is denoted as MaxF, and the minimum is denoted as MinF. Different face tracks are ranked in terms of their MaxF scores and only the top N_t tracks are selected as candidates for the following selection. The candidate tracks are then ranked in terms of their MinF scores, and the track with the largest MinF score is selected as the confident track. This selection process is graphically illustrated in Figure 5.2. With this mechanism, we aim to choose the track in which a certain frame is recognized as the "best match", and the rest frames are considered to be "good enough" matches. For a better understanding of this proposed mechanism, let's consider an extreme case where there are a large number of candidate tracks. For this case, we actually selects the most confident tracks by the averaged confidence scores of all the tracks. However, the selection results for this setting are possibly the tracks with minor between-frame variation. This may limit the generalization performance of the learned classifier. On the other hand, if N_t is too small, for example $N_t = 1$, it is quite likely to include false tracks especially when the initial classifier is trained on a small labeled set. Considering the total number of video tracks (around 2700) in our experi-

Algorithm 1 Framework of Adaptive Learning.

Input:
Initial Labeled Set $\mathcal{L}_o$, Related Set $\mathcal{L}_r = \emptyset$, Unlabeled Set $\mathcal{U}$, number of classes N, maximal iteration number N_{iter}, and N_t for TOP-N_t setting.

Output: Final Classifier $F = \{f_1 \ldots, f_N\}$

```
for i = 1 : N_iter do
    L ← L_o ∪ L_r
    Train classifier F^(i) = {f_1^(i), ..., f_N^(i)} on L ∪ U
    Compute g_k(x_j), ∀x_j ∈ U, k = {1, ..., N}
    for k = 1 : N do
        Compute MaxF for each track.
        Choose top N_t tracks as candidates according to MaxF.
        Select track T_p with the largest MinF from N_t candidates
        Set labels for x_j ∈ T_p as k
        L_r ← L_r ∪ {x_j ∈ T_p}
        U ← U \ {x_j ∈ T_p}
    end for
end for
```

ments, N_t is empirically set as 5 in the experiments. This small value of N_t may achieve a good trade-off between the diversity of the chosen tracks and the selection accuracy. The framework of adaptive learning based on such a selection strategy is described in Algorithm 1.

In general, Adaptive Learning (AL) is more robust to various changes in terms of pose, facial expression and so forth. Unlike traditional Semi-Supervised Learning, confident samples in Adaptive Learning obtain much higher influence than the remaining unlabeled samples in the next iteration of training. With the introduced video constraint, the labels are propagated from confident frames to those frames that are difficult to label based on information from the limited initial image seeds. The promoted diffident frames usually contain faces with more variations compared with the initial labeled samples. As a result, the classifier is trained with enriched "labeled data" with high diversity, and thus gains improvement on its generalization performance.

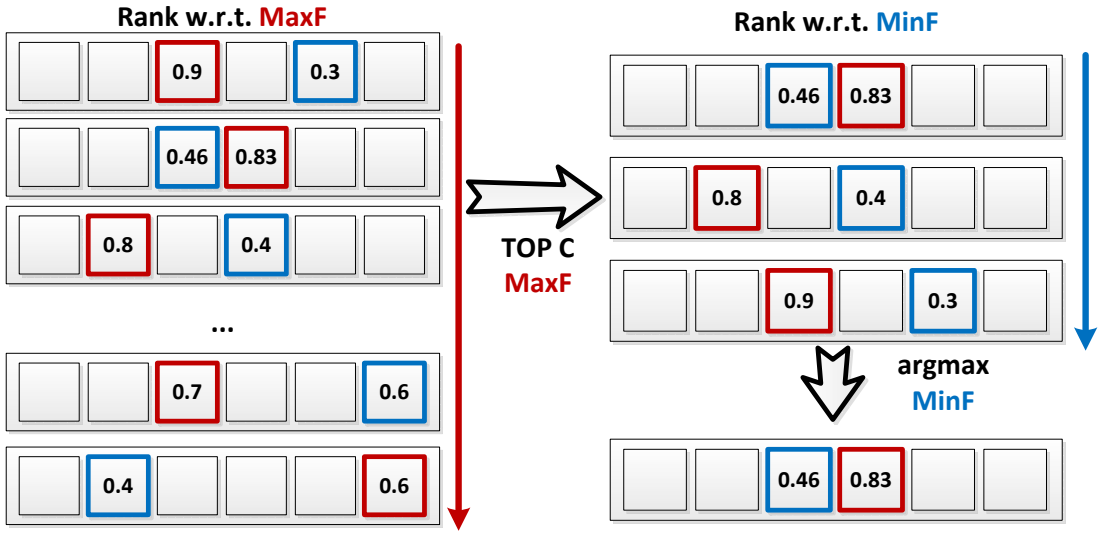

Figure 5.2: Illustration of confident tracks selection mechanism. Each large block represents a face track. The small red block refers to the most confident track and the blue block refers to the least confident track. Their corresponding confidence scores are shown inside. The first selection step (left) is based on MaxF and the second step (right) is based on MinF.

5.3 Adaptive Learning with Related Samples

The aforementioned straightforward adaptive approach simply treats *related samples* exactly the same as labeled samples in $\mathcal{L}_o$. Such an approach only works in the ideal case where no errors occur in selecting the confident tracks. However, selection errors are generally inevitable for the following two reasons: 1) poor discriminative capability of the learned classifier in the initial learning stage where the classifier is trained only with a small number of labeled images; 2) high similarity between different persons in certain frames. The errors in the selection process will cause semantic drifting [Shrivastava et al., 2012] and degrade the performance of the classifier. To address this problem, we introduce the concept of *related samples*, which is a comprise between *labeled* and *unlabeled* samples. Selected related samples are given higher weights than the remaining

unlabeled samples but lower weights than initial labeled samples in training the classifier. As a result, the initial accurately labeled data still contribute most to the learning process such that the undesired semantic drifting effect brought by promoting *related samples* is alleviated in a controlled manner. In the following subsections, we firstly review the Laplacian Support Vector Machine (LapSVM), and then introduce the proposed related LapSVM, which integrates the concept of adaptive learning and related samples in a unified framework.

5.3.1 Review of LapSVM

The aforementioned idea is formulated under the generalized manifold learning framework. In particular, we adopt Laplacian SVM (LapSVM), introduced by Belkin et al. [Belkin et al., 2006], as a concrete classifier learning method in this chapter.

LapSVM is a graph-based semi-supervised learning method. A sample affinity graph is denoted as $\mathcal{G} = \{V, E\}$, where V represents the set of nodes (data samples) and E refers to edges whose weights specify pair-wise similarity defined as follows

$$s_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2), \tag{5.2}$$

where σ is a parameter controlling the similarity based on sample Euclidean distance and is determined via cross-validation in this work.

In LapSVM, classifier f is learned by minimizing the following objective function:

$$\mathcal{J}(f) = \sum_{i=1}^{l} \max(1 - y_i f(\mathbf{x}_i), 0) + \gamma_A \|f\|_A^2 + \gamma_I \|f\|_I^2, \tag{5.3}$$

where $\|f\|_A^2$ represents the regularization in corresponding Reproducing Kernel Hilbert Space (RKHS) to avoid over-fitting. $\|f\|_I^2$ embodies the smoothness assumption on the underlying manifold, i.e., samples with high similarity have similar classifier responses. Here, we adopt a graph-based manifold regularizer as $\|f\|_I^2 = \sum_{i,j} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 s_{ij}$.

By defining the classifier in the RKHS according to the representer theorem [Scholkopf et al., 2001], we have the following classifier representation:

$$f(\cdot) = \sum_{i=1}^{l+u} \alpha_i k(\mathbf{x}_i, \cdot), \tag{5.4}$$

where $k(\cdot,\cdot)$ is a kernel function in RKHS. In this work, we adopt linear kernel trading-off the performance and computational complexity, i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$.

By substituting Eqn. (5.4) back into Eqn. (5.3), the objective function is equivalently rewritten as

$$\mathcal{J}(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \max(1 - y_i f(\mathbf{x}_i), 0) + \gamma_A \boldsymbol{\alpha}^T \mathbf{K}\boldsymbol{\alpha} + \gamma_I \boldsymbol{\alpha}^T \mathbf{K}\mathbf{L}\mathbf{K}\boldsymbol{\alpha} \tag{5.5}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_n\}^T$, and $\mathbf{K}$ is the n by n gram matrix over labeled and unlabeled sample points. $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is the laplacian matrix on the adjacency graph $\mathcal{G}$, where $\mathbf{D}$ is diagonal matrix with $d_{ii} = \sum_j s_{ij}$ and $\mathbf{S}$ is the weight matrix defined in Eqn.(5.2).

LapSVM can be directly applied in our adaptive learning framework. However, as pointed out before, the cumulative error in labeling the unlabeled data may cause the problem of semantic drifting. In the following subsection, we introduce the proposed related LapSVM to solve the problem.

5.3.2 Related LapSVM

Intuitively, to solve the problem of incorrect sample selection, the influence of selected samples should be more significant than the remaining unlabeled samples, but not greater than initial original labeled samples. Referring to LapSVM [Belkin et al., 2006], labeled data are prone to be the support vectors, or in other words, lying on the ± 1 margin, while there is no such constraint on unlabeled data. Selected frames, however, should lie between the decision boundary (uncertain unlabeled data) and the margin (labeled data). By considering the hard constraint in the video, frames from the same

track should be put on the same half-space with regard to the classifier decision boundary, as shown in Figure 5.3. These selected samples are treated as *related samples*, lying between the labeled and unlabeled samples.

We propose Related LapSVM to incorporate the concept of *related sample* into LapSVM. Formally, via introducing a weight ρ for the related samples in deciding the classifier boundary, the objective function of LapSVM in Eqn. (5.5) is changed into:

$$\begin{aligned} & \mathcal{J}(\boldsymbol{\varepsilon}, \boldsymbol{\alpha}) = \sum_{i=1}^{l} \varepsilon_i + \gamma_A \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} + \gamma_I \boldsymbol{\alpha}^T \mathbf{K} \mathbf{L} \mathbf{K} \boldsymbol{\alpha} \\ \text{s.t.} \quad & y_i \left(\sum_{j=1}^{l+u} \alpha_j k(\mathbf{x}_j, \mathbf{x}_i) + b \right) \geqslant 1 - \varepsilon_i, \forall \mathbf{x}_i \in \mathcal{L}_o \\ & y_{\mathcal{T}}^i \left(\sum_{j=1}^{l+u} \alpha_j k(\mathbf{x}_j, \mathbf{x}_i) + b \right) \geqslant (\rho \cdot \mathcal{C}_{\mathcal{T}}^i) - \varepsilon_i, \forall \mathbf{x}_i \in \mathcal{L}_r \\ & \varepsilon_i \geqslant 0, \forall \mathbf{x}_i \in \mathcal{L}, \, 0 \leq \rho \leq 1, \end{aligned} \tag{5.6}$$

where ε_i is the slack variable for $\mathbf{x}_i$. There are also a few works [Tzelepis et al., 2013, Tzelepis et al., 2015] that realize the idea of related samples. In these papers, the contributions of related samples are achieved via re-weighting the slack variables ε. The main motivation is to offer higher tolerance for the classification errors with regard to related samples during training. This chapter, in contrast, aims to lower the influence of related samples such that the original labeled samples are always dominant during training. Accordingly, we re-weight the margins instead of slack variables to suppress the effect of semantic drifting. Actually, the re-weighting for slack variables and margins are not contradictory, and thus can be incorporated into the LapSVM objective simultaneously. This chapter focuses more on solving the semantic drifting issue, and thus does not consider from the perspective of slack variables in order to avoid ambiguous interpretation of the experimental results.

The predicted label $y_{\mathcal{T}}^i$ and confidence score $\mathcal{C}_{\mathcal{T}}^i$ for the most confident frame in track $\mathcal{T}_i$ are defined as follows:

$$\begin{aligned} \mathcal{C}_{\mathcal{T}}^{i} &= g(\mathbf{x}_j), \\ y_{\mathcal{T}}^{i} &= \mathrm{sgn}(f(\mathbf{x}_j)), \end{aligned} \tag{5.7}$$

where $j = \arg\max_k g(\mathbf{x}_k), \forall \mathbf{x}_k \in \mathcal{T}_i$. In the two equations above, $g(\cdot)$ is the softmax function for calculating the confidence score. With Eqn. (5.7), each face track is tagged with the same label as the most confident sample within.

As shown in Eqn. (5.6), each related sample $\mathbf{x}_i \in \mathcal{L}_r$ is placed on a hyperplane with a distance $\rho \cdot \mathcal{C}_{\mathcal{T}}^{i}$ to the decision boundary. The further the hyperplane lies away from the decision boundary, the greater influence the related samples lying on it will have in determining the decision boundary. The underlying assumption is that the track with the sample of a higher confidence score has a higher probability to be the correct track, and thus should contribute more to constraining the learning process. The constraint in Eqn. (5.6) guarantees that the influence of a certain related sample is proportional to the corresponding confidence score. Also, a slack variable is imposed for each related sample, similar to the soft-margin concept in traditional SVM. ρ is a parameter in the range $[0, 1]$ to control the upper bound of the margin for related samples. A larger ρ indicates a stronger constraint on *related samples*. When ρ is set to 0, we only require all the frames within the same track to lie on the same half-space of the decision boundary.

Following the similar optimization method in [Belkin et al., 2006], the problem in Eqn. (5.6) can be written in the following Lagrange form,

$$\begin{aligned} \mathcal{J}_g(\boldsymbol{\alpha}, \boldsymbol{\varepsilon}, b, \boldsymbol{\beta}, \lambda) &= \sum_{i=1}^{l} \varepsilon_i + \gamma_A \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} + \gamma_I \boldsymbol{\alpha}^T \mathbf{K} \mathbf{L} \mathbf{K} \boldsymbol{\alpha} \\ &- \sum_{\forall i, \mathbf{x}_i \in \mathcal{L}_o} \beta_i (y_i (\sum_{j=1}^{l+u} \alpha_j k(\mathbf{x}_j, \mathbf{x}_i) + b) - 1 + \varepsilon_i) - \sum_{i=1}^{l} \lambda_i \varepsilon_i \\ &- \sum_{\forall i, \mathbf{x}_i \in \mathcal{L}_r} \beta_i (y_{\mathcal{T}}^{i} (\sum_{j=1}^{l+u} \alpha_j k(\mathbf{x}_j, \mathbf{x}_i) + b) - \rho \cdot C_{\mathcal{T}}^{i} + \varepsilon_i) \end{aligned} \tag{5.8}$$

.

According to the KKT conditions, we set the derivatives of L_g in terms of b and ε_i as zeros, which yields

$$\begin{aligned} \frac{\partial \mathcal{J}_g}{\partial b} = 0 \Rightarrow \sum_{i,\mathbf{x}_i \in \mathcal{L}_o} \beta_i y_i + \sum_{i,\mathbf{x}_i \in \mathcal{L}_r} \beta_i y_{\mathcal{T}}^i = 0, \\ \frac{\partial \mathcal{J}_g}{\partial \varepsilon_i} = 0 \Rightarrow 1 - \beta_i - \lambda_i = 0 \Rightarrow 0 \leqslant \beta_i \leqslant 1. \end{aligned} \tag{5.9}$$

By substituting Eqn. (5.9) into Eqn. (5.8) and canceling b, λ, ε, the lagrangian function becomes

$$\begin{aligned} &\mathcal{J}_g(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \gamma_A \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} + \gamma_I \boldsymbol{\alpha}^T \mathbf{K} \mathbf{L} \mathbf{K} \boldsymbol{\alpha} \\ &- \boldsymbol{\alpha}^T \mathbf{K} \mathbf{J}_L^T \mathbf{Y} \boldsymbol{\beta} + \sum_{\forall i, \mathbf{x}_i \in \mathcal{L}_o} \beta_i + \sum_{\forall i, \mathbf{x}_i \in \mathcal{L}_r} (\rho \cdot C_{\mathcal{T}}^i) \beta_i \\ &\text{s.t.} \quad 0 \leq \beta_i \leq 1, \forall \mathbf{x}_i \in \mathcal{L}_o \cup \mathcal{L}_r. \end{aligned} \tag{5.10}$$

Here $\mathbf{Y}$ is a diagonal labeled matrix, whose non-zero entries are set as label y_i for samples in $\mathcal{L}_o$ or predicted label $y_{\mathcal{T}}^i$ for samples in $\mathcal{L}_r$; we also define $\mathbf{J}_L = [\mathbf{I} \quad \mathbf{0}]$ where $\mathbf{I}$ is an identity matrix with a size equal to the cardinality of set $||\mathcal{L}||$.

Applying the KKT conditions again, we represent α by β:

$$\frac{\partial \mathcal{J}_g}{\partial \boldsymbol{\alpha}} = 0 \rightarrow \boldsymbol{\alpha} = (2\gamma_A \mathbf{I} + 2\gamma_I \mathbf{L} \mathbf{K})^{-1} \mathbf{J}_L^T \mathbf{Y} \boldsymbol{\beta}, \tag{5.11}$$

and $\mathbf{K}$ is invertible since it is positive semi-definite.

Finally, the corresponding dual form of Eqn. (5.6) can be rewritten as follows

$$\begin{aligned} \max_{\beta} \quad & \sum_{\forall i, \mathbf{x}_i \in \mathcal{L}_o} \beta_i + \sum_{\forall i, \mathbf{x}_i \in \mathcal{L}_r} (\rho \cdot C_{\mathcal{T}}^i) \beta_i - \frac{1}{2} \boldsymbol{\beta}^T \mathbf{Q} \boldsymbol{\beta}, \\ \text{s.t.} \quad & \sum_{\forall i, \mathbf{x}_i \in \mathcal{L}_o} \beta_i y_i + \sum_{\forall i, \mathbf{x}_i \in \mathcal{L}_r} \beta_i y_{\mathcal{T}}^i = 0, \\ & 0 \leqslant \beta_i \leqslant 1, \end{aligned} \tag{5.12}$$

where

$$\mathbf{Q} = \mathbf{Y} \mathbf{J}_L \mathbf{K} (2\gamma_A \mathbf{I} + 2\gamma_I \mathbf{L} \mathbf{K})^{-1} \mathbf{J}_L^T \mathbf{Y}. \tag{5.13}$$

Eqn. (5.12) is a standard QP problem. The optimal solution can be derived utilizing traditional off-the-shelf SVM QP solvers, and we use SPM:QPC solver[1] in this chapter.

[1] http://sigpromu.org/quadprog/

5.3.3 Classification Error Bound of Related LapSVM

This section provides a theoretical classification error bound for the proposed related LapSVM, via comparing with the established error bound of standard LapSVM. The quantitative evaluation of related LapSVM is given in details in the experiment section.

Given a data distribution $\mathcal{D}$ and classifier function class $\mathcal{F}$, the classification error of LapSVM is bounded by the summation of the empirical error, function complexity and data complexity, as formally stated in the following lemma [Sun, 2011].

Lemma 1 *([Sun, 2011]). Fix $\delta \in (0,1)$ and let $\mathcal{F}$ be a class of functions mapping from an input space $\mathcal{X}$ to $[0,1]$. Let $\{\mathbf{x}_i\}_{i=1}^{l}$ be drawn independently according to a probability distribution $\mathcal{D}$. Then with probability at leat $1-\delta$ over random draws of samples of size l, every $f \in \mathcal{F}$ satisfies*

$$E_{\mathcal{D}}[f(\mathbf{x})] \leq \hat{E}[f(\mathbf{x})] + R_l(\mathcal{F}) + \sqrt{\frac{\ln(2/\delta)}{2l}}, \tag{5.14}$$

where $\hat{E}[f(\mathbf{x})]$ is the empirical error averaged on the l examples and $R_l(\mathcal{F})$ denotes the Rademacher complexity of the function class $\mathcal{F}$.

By utilizing the error bound of SVM [Cristianini and Shawe-Taylor, 2000], $\hat{E}[f(z)] \leq O\left(\|\xi\|_2^2 \log^2 l\right)$, we can further bound the error of LapSVM in terms of the slack variable ε_i as follows,

$$E_D[f(\mathbf{x})] \leq O\left(\sum_i \varepsilon_i^2 \log^2 l\right) + R_l(\mathcal{F}) + \sqrt{\frac{ln(2/\delta)}{2l}},$$

where ε_i is the slack variable for sample $\mathbf{x}_i$ in the labeled or related sample set. The proposed related LapSVM reduces the classification error bound over LapSVM via properly re-weighting the slack variable for the unconfident/noisy samples. Specifically, consider the case where a sample $\mathbf{x}_j$ is selected as a confident sample but labeled incorrectly. For $\mathbf{x}_j$, training the classifier actually minimizes an incorrect slack variable ε_j, and maximizes the correct slack variable $\hat{\varepsilon}_j = 1 - \varepsilon_j$, due to its opposite label. $\hat{\varepsilon}_j$ is

maximized within the range of $[0, 2]$. Thus, the error bound is increased to

$$E_D[f(z)] \leq O\left((\sum_{i \neq j} \varepsilon_i^2 + \hat{\varepsilon}_j^2) \log^2 l\right) + R_l(\mathcal{F}) + \sqrt{\frac{ln(2/\delta)}{2l}}.$$

In contrast, related LapSVM reduces the feasible range of $\hat{\varepsilon}_j$ to $[0, 2 - \rho \cdot \mathcal{C}_{\mathcal{T}}^j]$. Consequently, the value of $\hat{\varepsilon}_j$ is decreased, and related LapSVM has lower error bound than standard LapSVM, i.e.,

$$E_D[f_{\text{re-LapSVM}}(x)] \leq E_D[f_{\text{LapSVM}}(x)]. \tag{5.15}$$

The above analysis can be generalized to the case where more unlabeled samples are labeled incorrectly. Thus we can conclude that the related LapSVM reduces error bound via handling the incorrectly labeled samples better.

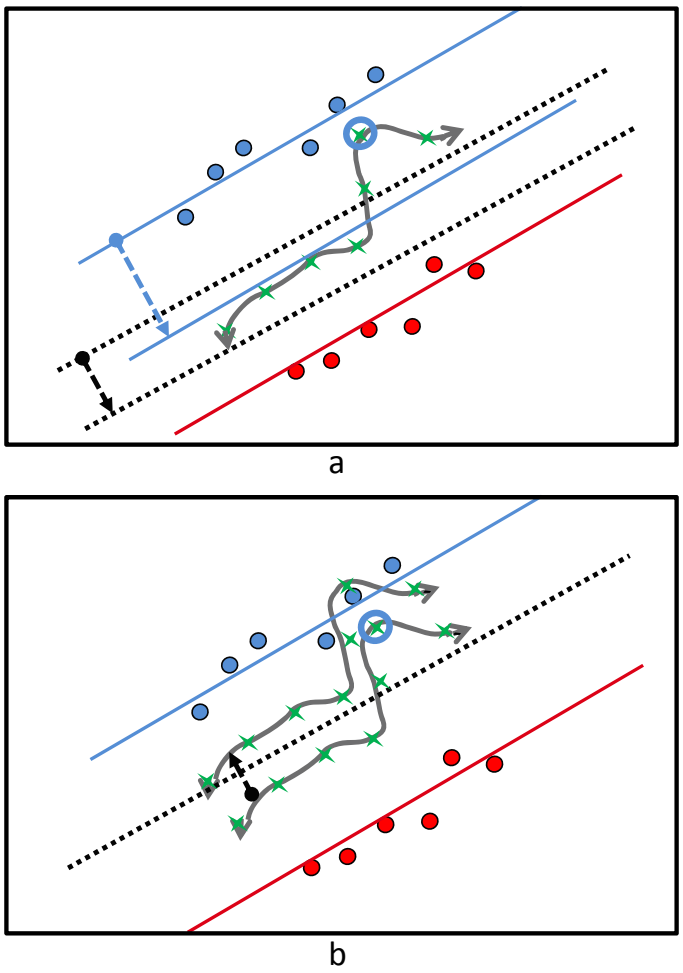

Figure 5.3: Illustration on naive Adaptive Learning and Related LapSVM. Blue and red dots represent labeled samples for positive and negative class, respectively. Green stars represent face frames in a face track (gray curve). A certain frame (star in blue circle) is recognized as the most confident sample with a positive predicted label. Block (a) shows the change of margin (blue and red line) and decision boundary (black dashed line), as indicated by the colored arrows, for naive Adaptive Learning. Block (b) shows the change after including the concept of *related sample*. For naive adaptive learning, the margin is completely determined by selected samples, i.e., the initial labeled images are unable to constrain the learning process. However, for Related LapSVM, the influence of *related samples* do not overtake the original labeled set and the margin is retained as desired.

5.4 Experiments

We conduct extensive experiments to evaluate the effectiveness of the proposed adaptive learning method for celebrity identification. This section is organized as follows. Subsection 5.4.1 introduces the details of construction of the used database. We demonstrate the experimental settings in details in subsection 5.4.2. Subsection 5.4.3 shows a naive approach of including the video constraint in building the sample affinity graph and demonstrates that video context can improve the performance with a limited degree. Subsection 5.4.4 and 5.4.5 show the effectiveness of related samples in both supervised and semi-supervised learning scenarios. The average precision is reported on both image and video testing set. Subsection 5.4.6 illustrates the performance curve of the proposed method along with learning iterations. We also include in the last subsection experiments of related samples on a public database - YouTube Celebrities Database.

5.4.1 Database Construction

Since there are rare databases with sufficient image and video samples for celebrity identification, in this chapter, we construct a database for benchmarking different methods for this task. The collection of image and video data is described as follows.

Image Data

We select 30 celebrities who are well-known within their fields so that sufficient corresponding video data can be crawled. For each individual, we retrieve about 100 clear images from Google Image using the names of celebrities as queries. We manually label all the images and mark the locations of eyes. All faces are then normalized via a standard affine transformation. There is not any strict constraint in photography conditions – different poses, facial expressions and illumination conditions are all allowed.

15 images are randomly sampled to form the training image set, while the remaining are used as the testing set. We report the average precision (AP) on 5 different random training-testing splits. The list of celebrities chosen in the database is given in Table 5.1.

Video Data

Querying with the celebrity names, a video corpus consisting of about 300 video clips is downloaded from video sharing websites, e.g., YouTube. Note that for the following experiments, we assume that the videos are unlabeled for the following reasons: a) the keyword searching results are not reliable, and videos are not necessarily related with the celebrities; b) there may also be other individuals other than the celebrities of interest in the returned videos.

Only the detected face tracks are considered in the iterative adaptive learning process. Thus based on the video constraint, the label is transfered from confident frames to uncertain frames within the same track. Besides, by only considering the detected tracks, the volume of frames that need to be processed can be largely reduced to accelerate the learning process. To obtain reliable face tracks, a robust foreground correspondence tracker [Wang et al., 2011] is applied for each shot.

Here video shot segmentations are automatically detected with the accelerating shot boundary detection method [Gao and Ma, 2011]. More specifically, the Focus Region (FR) in each frame is defined, and using a skip interval of 40 frames, the method not only speeds up the detection process, but also finds more subtle transitions.

After segmenting the video into shots, the tracking process takes the results of OKAO face detection[1] as input, and generates several face tracks using the tracking algorithm in [Wang et al., 2011]. The face tracks are then further pruned via fine analysis of faces as follows:

[1] http://www.omron.com/r_d/coretech/vision/okao.html

Occupation	Name	Gender	Video Source
Politician	Barack Obama	M	Speech News Report
	Yingjiu Ma	M	
	Al Sharpton	M	
Western Actor	Adam Sandler	M	Movies Interviews
	Alexander Skargard	M	
	Alan Alda	M	
	Anthony Hopkins	M	
	Alan Rickman	M	
	Alan Tricke	M	
	Amy Poehler	F	
	Alicia Silverstone	F	
Asian Actor	Chao Deng	M	Movies Interviews
	Baoqiang Wang	M	
	Zidan Zhen	M	
	Benshan Zhao	M	
	Bingbing Fan	F	
	Wei Tang	F	
	Yuanyuan Gao	F	
Singer	Dehua Liu	M	Music Albums Concerts
	Katty Perry	F	
	Wenwei Mo	F	
	Xiaochun Chen	M	
	Yanzi Sun	F	
Hoster &Anchor	Anderson Cooper	M	News Report Talk Show TV Programs
	Fujian Bi	M	
	Lan Yang	F	
	Jing Chai	F	
CEO	Yun Ma	M	Commercial News Product Launch Video
	Bill Gates	M	
	Steve Jobs	M	

Table 5.1: Celebrities included. We choose people with different occupations as listed above. For different occupations, video data are collected from different video sources correspondingly.

- Duration. Short tracks with less than 30 frames are discarded, since these tracks are often introduced by false positive detections.

- Clusters. K-means clustering is applied on each track, and only those frames clos-

est to clustering centers are chosen as corresponding representative faces.

Finally we acquire around 2,700 video tracks in total with nearly 90 tracks per individual.

Feature for Face Recognition

We adopt the following three popular hand-crafted features in face recognition – Gabor, LBP and SIFT feature. Details of the feature extraction are listed below:

- **Gabor Feature**. Gabor filter [Daugman, 1985] has been widely used for facial feature extraction due to its capability of capturing salient visual properties, such as spatial localization, orientation selectivity as well as spatial frequency characteristics. In this section, we adopt a common setting for extracting gabor feature: wavelet filter bank with 5 scales and 8 orientations, central frequency is set as $\sqrt{2}$, and filter window width is set as 2π.

- **Local Binary Patter Feature**. LBP captures the contrast information of the central pixel and its neighbors. The advantage of LBP lies in its robustness to illumination and pose variations. We use a variant of LBP - multi-block LBP [Ojala et al., 2002b]. In the feature selection, the image is firstly segmented into several blocks to keep a certain amount of geometric information. Each face image is divided into 5×4 sub-regions and then for each sub-region uniform patterns are extracted and concatenated as bins for a histogram representation.

- **SIFT Feature**. A nine-point SIFT feature is used in the experiments. Referring to the work of Everingham et al. [Everingham et al., 2006], a generative model is adopted to locate the nine facial key-points in the detected face region, including the left and right corners of each eye, the two nostrils and the tip of the nose and the left and right corners of the mouth followed by 128-dim SIFT feature [Lowe and G, 1999] extraction process.

The vectors of the above three features are normalized individually by ℓ_2-norm and concatenated into a single super-vector for each image/frame.

5.4.2 Experiment Settings

In the following experiments, the initial training image set is constructed by randomly sampling 15 images per person from the labeled image data, and the rest images are used for testing. We run this sampling process for 5 times in each experiment and report the mean precision in this section.

We consider two scenarios for experiments – 10-person and 30-person scenario. In the 10-person scenario, 10 celebrities are selected randomly from the name list in Table 5.1 and corresponding training samples are chosen as above. We perform such random selection processes for 3 times and then reported the average precision (AP). In the 30-person scenario, we use the training samples of all celebrities.

For AL, we follow the procedures in Algorithm 1 with the value of parameter N_t set as 5. The maximal iteration number is set as $N_{iter} = 15$, and the results for AL based approaches are reported as the accuracy of the final learning iteration. The parameter η in Eqn. (5.1) is set as 0.7. In Related LapSVM defined in Eqn. (5.6), γ_I and γ_A are set as 10^{-2} and 1 respectively, and ρ is empirically set as 0.3.

5.4.3 Video Constraint in Graph

LapSVM [Melacci and Belkin, 2011] is a graph-based classifier and we take a simple extension to incorporate the video context information into LapSVM framework as a baseline.

The general idea is to include the video constraint when constructing the affinity matrix, which is composed of the similarities among training instances. A naive approach

is to set the similarity of frames from the same track to be 1. Nevertheless, experiments show that this setting usually results in a degradation in performance. A possible reason could be that the weight among consecutive frames becomes much larger than other entries within the weight matrix, which makes the classifier dominated by the constraints on corresponding samples other than labeled instances. Therefore, to ensure the balance of sample weights, the weight is defined as the summation of graphic similarity and video constraint. In detail, the edge between consecutive frames is defined as,

$$\begin{aligned} s_{ij} = \lambda \cdot \exp\{-(\mathbf{x}_i - \mathbf{x}_j)^2 / 2\sigma^2\} \\ + (1-\lambda) \cdot \min\{\zeta \cdot \mu_W, 1\}, \end{aligned} \tag{5.16}$$

where μ_W is the mean of matrix $\mathbf{S}$.

In Eqn. (5.16), we confine $\lambda \in [0, 1]$ and $\zeta \in [1, 10]$ empirically. We tune the values for λ with a step of 0.1 and ζ with a step of 1 within their corresponding ranges via cross-validation. Experiments show a small improvement over LapSVM of 1% on average. This approach is named as Lap+V, and is taken as the baseline algorithm in the following experiments.

5.4.4 Related Sample in Supervised Learning

In this subsection, we evalute the effect of *related sample* on SVM. γ_I in Eqn. (5.6) is set as 0 and the classifier is defined only in terms of labeled samples, i.e., $f(\cdot) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \cdot)$. We compare the following methods – SVM, ST-SVM (self-training with SVM), AL-SVM (adaptive learning with video constraint), and Re-SVM (related SVM). All these methods are evaluated on both image and video data in 10-person and 30-person scenarios respectively. Average precision is reported in Table 5.2 and 5.3 with varying numbers of labeled training images.

		3	5	7	10	12	15
Image	Lap+V	**50.83**	**60**	**68.33**	76.67	82.5	84.17
	SVM	39.17	48.75	58.75	75	75.83	78.75
	ST-SVM	41.67	51.25	61.25	75.42	76.25	80.83
	AL-SVM	43.34	51.25	58.75	77.92	80.84	82.09
	Re-SVM	50.42	54.17	65.42	**82.5**	**82.92**	**84.59**
Video	Lap+V	**53.09**	48.38	49.97	56.62	70.32	73.41
	SVM	30.4	33.09	45.44	62.23	66.76	70.68
	ST-SVM	29.11	34.17	44.21	64.12	66.88	72.93
	AL-SVM	46.32	50.3	45.84	55.73	63.37	75.42
	Re-SVM	49.55	**50.45**	**50.57**	**76.26**	**79.26**	**78.18**

Table 5.2: Comparison on the average precision (%) of different SVM based methods in the 10-person scenario.

		3	5	7	10	12	15
Image	Lap+V	**48.87**	**62.83**	**72**	**82**	**84.5**	**86.5**
	SVM	39.33	51.17	61.67	75	79.5	82.17
	ST-SVM	39.33	50.33	60.5	75.33	79	81.67
	AL-SVM	38.34	48.17	57.34	72.17	79	84.5
	Re-SVM	41.5	52.84	63	76.34	82.5	84.34
Video	Lap+V	42.13	46.48	45.93	56.61	62.39	60.82
	SVM	31.97	38.79	41.41	50.88	60.48	64.95
	ST-SVM	32.26	38.08	40.74	51.55	59.67	63.61
	AL-SVM	36.06	38.35	48.95	64.01	69.55	74.02
	Re-SVM	**49.28**	**46.67**	**49.09**	**67.03**	**71.47**	**75.42**

Table 5.3: Comparison on the average precision (%) of different SVM based methods in the 30-person scenario.

For traditional self-training, only those frames with high similarity to the initial training samples are selected to enlarge the training set. Thus, the variations in the selection samples are limited. Limited number of labeled samples may decrease AP due to the high error rate during selection, while, more labeled samples usually result in improve-

ment for ST-SVM. However, the difference for either degradation and improvement is very small: less than 1%. Straight-forward adaptive learning (AL+SVM) demonstrates similar performance, but the range for both degradation and improvement are largely increased to around 4%.

Related SVM adjusts the margin for each sample in accordance with their confidence scores, such that we can amplify the positive influence of more confident samples while suppressing the negative influence of less confident samples. Generally, by regarding selected samples as related samples, the classifier is much more robust to selection errors. As shown in Table 5.2 and 5.3, the improvement of Re-SVM over SVM is around 5% on image data and 12% on video data. In most cases where the number of labeled samples is small (e.g. the number is 3 or 5), the initial classifier is unreliable. Normally, around half of the selected tracks are not correctly labeled by the classifier. Related SVM can significantly degrade the impact of error tracks and provide considerable AP improvement. With sufficient labeled training samples (e.g. 12 or 15), the generalization performance of the classifier is significantly improved. The error rate in selecting tracks is low, and thus correct samples play a dominant role in training. In such a case, the improvement brought by related samples becomes less significant.

Note that there is still an considerable performance gap between Related SVM and LapSVM with video constraint (Lap+V) on the image testing dataset: 3% and 6% in 10-person and 30-person cases. A possible reason lies in the fact that both training and testing samples are static images downloaded from Google. The correlation between video data and image data is low. As a consequence, the right tracks selected in AL will result in minor improvement for testing on images, while, the incorrect tracks will degrade the performance to a certain extent. The impact of error tracks is relatively significant compared with the influence of the right tracks. However, on the video dataset, Related SVM outperforms LapSVM with a margin of 5% and 7% in both 10-person and 30-person cases. Especially, when sufficient labeled samples are fed into the training

		3	5	7	10	12	15
Image	Lap+V	50.83	60	68.33	76.67	82.5	84.17
	ST-LapSVM	48.75	56.67	66.25	77.5	78.75	82.08
	AL-LapSVM	**50.84**	46.67	73.34	81.25	82.92	87.92
	Re-LapSVM	49.17	**61.67**	**75.84**	**83.34**	**85.42**	**88.34**
Video	Lap+V	53.09	48.38	49.97	56.62	70.32	73.41
	ST-LapSVM	34.83	42.39	48.62	66.34	71.19	75.72
	AL-LapSVM	48.21	16.67	39.27	64.06	63.76	79.83
	Re-LapSVM	**53.46**	**55.28**	**77.01**	**83.13**	**83.28**	**84.36**

Table 5.4: Comparison on the average precision (%) of different LapSVM based methods in the 10-person scenario.

		3	5	7	10	12	15
Image	Lap+V	48.87	62.83	72	82	84.5	86.5
	ST-LapSVM	45.33	60	72.17	82.33	84.67	86.67
	AL-LapSVM	40.17	52.34	64.17	76.67	78.34	84
	Re-LapSVM	**49**	**64.67**	**72.84**	**83.84**	**84.67**	**87.5**
Video	Lap+V	42.13	46.48	45.93	56.61	62.39	60.82
	ST-LapSVM	38.22	49.5	59.18	**64.76**	70.38	69.71
	AL-LapSVM	26.41	41.52	39.75	57.32	62.68	61.75
	Re-LapSVM	**42.29**	**50.66**	**57.16**	63.33	**71.52**	**72.9**

Table 5.5: Comparison on the average precision (%) of different LapSVM based methods in the 30-person scenario.

process - 10 or more, the improvement can be up to 20%.

5.4.5 Related Samples in Semi-Supervised Learning

In this subsection, we examine the effect of related samples in Semi-Supervised Learning and take LapSVM and Transductive Support Vector Machine (TSVM) as the base classifiers for AL.

		3	5	7	10	12	15
Image	TSVM	41.67	51.67	60.42	75.83	77.08	79.58
	AL-TSVM	39.58	48.75	59.17	74.58	79.17	85
	Re-TSVM	**42.5**	**52.08**	**62.5**	**80**	**78.75**	**85**
Video	TSVM	32.01	38.82	46.91	61.87	68.11	71.94
	AL-TSVM	**39.24**	40.05	43.05	61	73.56	78.66
	Re-TSVM	36.06	**42.66**	**47.39**	**75.03**	**75.99**	**81.92**

Table 5.6: Comparison on the average precision (%) of different TSVM based methods in the 10-person scenario.

When building up the affinity graph in LapSVM, video constraint in Eqn. (5.16) is not included. Related LapSVM (Re-LapSVM) is considered as another way of incorporating video constraint into the learning process other than Lap+V in Section 5.4.3. The video context information is utilized in the process of promoting tracks into the related set.

We investigate whether further improvement of LapSVM can be brought by Re-LapSVM over Lap+V. The results are given in Table 5.4 and 5.5. We observe similar results in the comparisons among self-training with LapSVM (ST-LapSVM), straightforward Adaptive Learning (AL-LapSVM) and Adaptive Learning with related samples (Re-LapSVM). Re-LapSVM outperforms both ST-LapSVM and AL-LapSVM. Re-LapSVM demonstrates a better tolerance to selection errors than the AL-LapSVM, especially for cases with 3, 5 and 7 labeled samples. More importantly, the comparison between Lap+V and Re-LapSVM provides more insightful results. Re-LapSVM demonstrates a significant advantage over Lap+V. In particular, the enhancement on AP is around **4**% for 10-person image case, **1.5**% for 30-person image case, **16**% for 10-person video case, and **8**% for 30-person video case, respectively.

In the implementation of TSVM, we optimize TSVM following Collober et al. [Collobert et al., 2006] with Concave-Convex Procedure (CCCP). The objective function of TSVM is non-convex, and CCCP optimizes the problem by solving multiple quadratic

programming subproblems. For each QP subproblem, we follow the similar way of incorporating related samples as in Eqn. (5.6) in Sec. 5.3.2. Since the optimization of TSVM is slow, we only conduct experiments in 10-person scenario. The results are demonstrated in Table 5.6.

Clearly, the performance of TSVM is worse than that of LapSVM, especially when labeled samples are limited. However, the comparison between TSVM and LapSVM is out of the scope of this work. Here, our focus is on whether related samples improve the performance of TSVM as well. As shown in Table 5.6, Re-TSVM outperforms both TSVM and AL-TSVM with an improvement of around 3%.

5.4.6 Learning Curves of Adaptive Learning

In this subsection, we investigate the behaviors of different approaches by investigating the average precision with respect to the iteration number. In this experiment, the labeled set for testing and training is fixed for a fair comparison. The maximal iteration count is set as 15, and accuracy on testing data is reported for each iteration. Since the learning curve is similar for most simulation runs, Figure 5.4 illustrates one run for LapSVM-based Adaptive Learning.

It is easy to observe that straightforward Adaptive Learning (Naive AL) shows a noisy curve since it is quite sensitive to the selection errors. If the correct track is chosen, accuracy will demonstrate an obvious increase, and the performance will drop suddenly if errors occur in the process of selection. Re-LapSVM with $\rho = 0$ shows a smooth learning curve and converges. Re-LapSVM with $\rho > 0$ shares the similar behaviors of the two approaches to some extent: the trend of AP is increasing but with minor turbulence. The parameter ρ in Eqn. (5.6) is an important factor controlling the relative influence compared with the labeled image samples in the learning process. Larger ρ will render the learning curve closer to straightforward AL, while smaller ρ pushes the

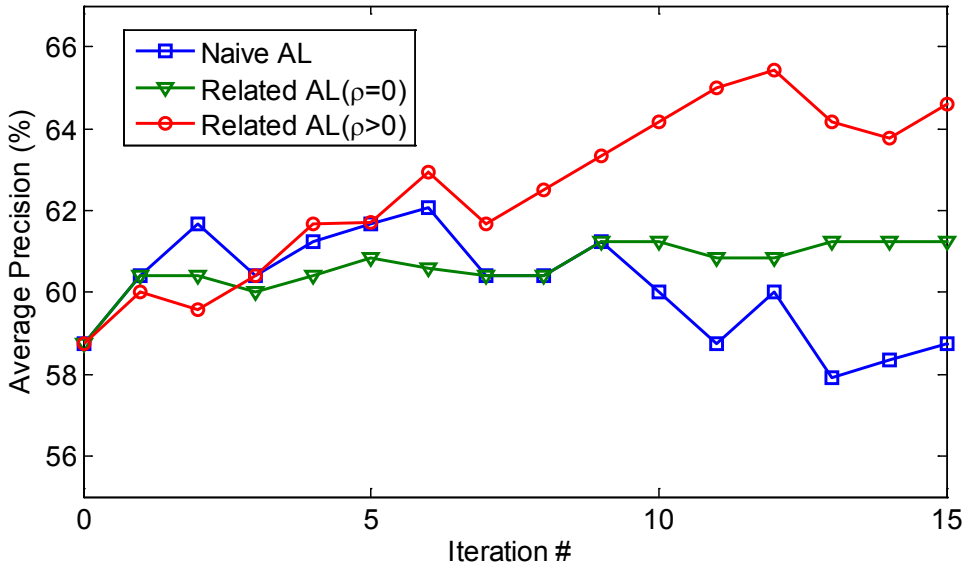

Figure 5.4: Learning Curves of three approaches: Naive AL, Related AL ($\rho = 0$) and Related AL ($\rho > 0$).

learning curve towards related AL with $\rho = 0$. An exemplar illustration of simulation results is also presented in Figure 5.5. In general, the observed results are consistent with our expectation.

5.4.7 YouTube Celebrity Dataset

We also evaluate the proposed algorithm on a public dataset – the YouTube Celebrity Dataset [Kim et al., 2008], which contains 1,910 sequences of 47 subjects. All the sequences are extracted from video clips downloaded from YouTube by evicting frames that do not contain celebrites of interest. Most of the videos are of low resolution and recorded at high compression rates. The size of frames ranges from 180×240 to 240×320 pixels.

Following the similar methods described in Section 5.4.1, face tracks are extracted within each video sequence. Only celebrities with more than 30 tracks are included

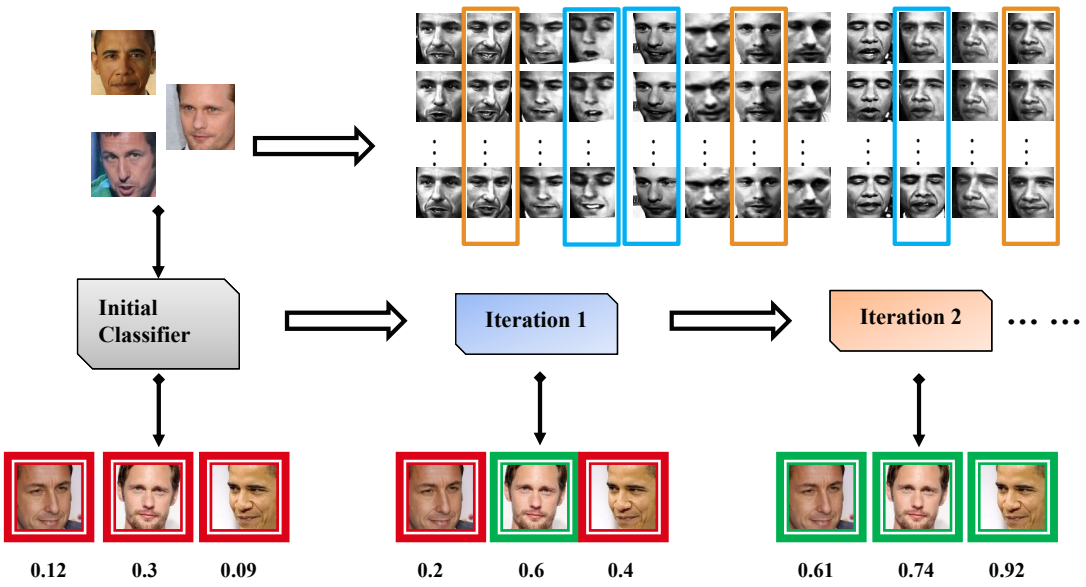

Figure 5.5: Examples of iterative improvement. The upper left static images are used for training the initial classifier, and the gray image matrix represents the pool of video tracks with each column standing for a track. In Iteration 1, tracks in blue bounding box are chosen, while in Iteration 2, tracks in orange bounding box are selected. The lowermost row are examples of testing images with corresponding confidence scores shown below. Red frame indicates wrong decision and green frame indicates right decision. With more tracks selected into the training pool, the confidence score on the testing dataset is rising.

in this experiment and the final number of identities is 32. Since there is no separate image set for the initial training stage as in our approach, we randomly sample 5 tracks for each celebrity. All the frames within are then treated as initial labeled samples. This sampling process is repeated for 5 times and the corresponding averaged results are shown in Table 5.7. The results are similar to those observed on our own dataset and the improvement of Re-LapSVM over Lap+V is around 4% on average.

Compared with the results on our own dataset, the improvement of Related LapSVM is less significant over the baseline algorithms. The reason is that the proposed method targets at solving a common problem in real applications, namely it is difficult to collect many training images to train reliable initial classifiers. When the number of la-

	3	5	7	10	12	15
SVM	45.73	47.3	55.05	57.66	62.25	41.33
ST-SVM	43.35	47.47	55.07	57.39	63.36	63.5
Lap+V	49.97	50.93	60.37	63.83	67.81	68.25
ST-LapSVM	51.3	52.3	61.55	63.92	68.05	68.29
AL-LapSVM	55.93	55.72	62.02	65.33	68.59	68.8
Re-LapSVM	**58**	**58.09**	**65.61**	**65.81**	**69.81**	**68.87**

Table 5.7: Comparison on the average precision (%) of different LapSVM based methods in Youtube Celebrities Database.

beled training images is small, the classifiers are not reliable, and errors in selecting the confident video tracks by such weak initial classifiers are inevitable. In this case, the performance of classifiers may degrade severely due to incorporating more and more noisy or incorrect samples. Thus, the improvement brought by related LapSVM is more significant with more noisy tracks selected.

Compared with the proposed dataset, the error rate in selecting confident tracks on Youtube dataset is much lower. Thus the performance gain of Re-LapSVM is smaller on the Youtube dataset compared with on our own dataset. The reasons of lower track selection error on the Youtube dataset are two-fold: 1) The Youtube dataset only contains videos, so we train the initial classifier using the video data. Such video-domain classifiers perform more accurately in selecting the confident remaining video tracks than the initial classifiers trained from image-domain in our own dataset; 2) The face sequences (tracks) for each individual in Youtube faces dataset are usually extracted from only 2-3 videos, and the correlation/similarity among different sequences from the same video is quite high. However, the dataset built in this work contains tracks from about 10 different videos for each celebrity. Thus, our video dataset is much more diverse and difficult for track selection than the Youtube face dataset. Due to the above two reasons, the performance improvement on Youtube dataset achieved by Re-LapSVM is less

significant than that on our dataset.

5.5 Conclusions

In this chapter, a novel adaptive learning framework was proposed for the celebrity identification problem inspired by the concept of "Baby learning". The classifier is initially trained on labeled static images, and gradually improves by augmenting confident face tracks into the knowledge base. We also proposed an effective approach that improves the robustness of classifiers to selection errors by assigning weak adaptive margin for those selected samples. Extensive experiments are conducted in both supervised and semi-supervised learning settings for the problem of celebrity identification. Experiment results on two databases show that the improvement on accuracy is significant and inspiring. Although in this chapter we only consider the task of celebrity identification, the proposed method is a general approach and can also be easily extended to other problems in computer vision, such as object detection, object recognition and action recognition.

Chapter 6

Conclusions and Future Work

This thesis targets at the challenging problem of multi-modal face recognition in the wild. The problem suffers from challenges such as large intra-class variations and scarcity of labeled samples. In this thesis, we propose two deep learning based methods and a semi-supervised framework to handle these challenges respectively. Each method is compared with the state-of-the-art approaches in extensive experiments, which show promising results on several benchmark databases. The specific content and achievements of each chapter can be summarized as follows.

Chapter 3 introduces a deep framework to handle the common local variations for face verification. The proposed framework consists of two major components. The first component is a Deep Mixture Model that aims at exploring the patch-wise correspondences. The second part is a composite framework that fuses multiple sub-nets trained on patches with different geometric positions and lighting conditions. The combination of these two networks leads to an effective patch-based feature representation for

face verification in the wild. Without relying on the hand-crafted features, the proposed method achieves an encouraging performance on two benchmark datasets.

In Chapter 4, we propose a conditional Convolutional Neural Network (c-CNN) to tackle the multi-modal face recognition in a more general way. In c-CNN, no prior knowledge is required on the modality distribution of either the whole dataset or individual sample. Instead, c-CNN learns the partition with regard to the inherent modalities and the corresponding modality-specific feature representation in a unified framework. Extensive experiments are conducted on multi-view and multi-occlusion face recognition problems respectively. Without modifications on the network structure, the proposed c-CNN method is able to automatically learn the modality-specific representation, and shows considerable improvements in both evaluation scenarios.

Chapter 5 incorporates the video context constraint into the Semi-Supervised Learning frameworks to tackle the problem of celebrity identification. The introduction of the video context helps to select the confident continuous inputs in order to gradually improve the recognition capability, which shares certain similarity with the learning process of human beings. Furthermore, the idea of adaptive margin for related sample is also proposed to address the common issue of semantic drifting. Experiments demonstrate that the video context information brings significant improvements over the standard self-training scheme but is sensitive to early errors in selection. Adaptive margin, on the other hand, results in a significant suppressing effect on the selection errors.

6.1 Relationships between Chapters

Each chapter introduces an individual work that seems to be separate from each other. However, the ideas in these chapters are not exclusive but connected from certain as-

pects.

To begin with, the methods presented in this thesis follows two lines of research corresponding to the two challenges mentioned in Section 1.1. There are two general ways to solve the deficiency of labeled samples. One option is to explore the crucial information from vast unlabeled samples to regularize the propagation of labels. This direction is studied in the branch of Semi-Supervised Learning, which is also the basis of Chapter 5. The other option is to collect and label a large number of face images manually. In recent years, manual efforts have been spend on creating such large database of labeled samples. The increasing amount of labeled training images is one of the major reasons for the rapid development of deep learning based methods. Therefore, the high discriminative capability of deep learning on representing the multi-modal face data can not be achieved without efforts in addressing the data-scarcity issue either.

Secondly, the method introduced in Chapter 5 does not have a specific requirement on the classifier. In fact, the adaptive learning method has already been evaluated with both supervised and semi-supervised classifiers in Section 5.4. For SSL based adaptive learning, the adaptation with different classifiers, including LapSVM and TSVM, is well examined in extensive experiments. Deep learning based approaches combine the feature learning and classifier learning in a joint manner, thus can be applied in the adaptive learning framework as classifiers naturally. Some drawbacks do exist for such direct integration. DNN may not perform well with a small number of training samples, but the introduction of video context has also shown encouraging results with a relatively large set of labeled samples. As for the adaptive margin, the cost defined in Eqn. (5.6) is differentiable, thus can be optimized by the standard back-propagation.

Finally, the patch-based representation can be integrated with the idea of c-CNN. The contributions of Chapter 3 are in two folds – part-based deep representation and Deep Mixture Model for facial patch correspondence. The part-based method takes

advantage of the complementary effects among different geometric parts to improve the diversity of learnt representation. The idea of c-CNN is to explore the diversity in terms of different modalities to partition the data. Therefore, the two methods are not contradictory conceptually. The proposed c-CNN can be adopted as a basic sub-net in Convolutional Fusion Network.

Moreover, both CFN and c-CNN are applied in face recognition with pose. C-CNN emphasizes more on the improvements for faces with a large pose span, e.g., -90 to $+90$ degree. CFN, on the other hand, sets focus on faces with relatively small poses, i.e., the near frontal faces. The difference with regard to modality is not severe enough, thus the modality-specific features of c-CNN may not be so distinct for different poses to further improve the performance.

6.2 Future Work

This thesis discusses both challenges for unconstrained face recognition, and proposes corresponding approaches to address these issues. However, there are still some limitations.

- The part-based facial representation is achieved with a two-stage deep framework in Chapter 3. DMM and CFN are optimized with regard to two different objectives – DMM aims to maximize the posterior possibility of the mixture model; CFN targets at reducing the verification errors of the training pairs directly. Moreover, the learning processes of the two components are separate, which means that the learnt representation may not be optimal for the given problem.

- C-CNN is realized via integration of decision tree and CNN in Chapter 4. In current approach, the convolution kernels in each layer are allocated to all the tree

nodes equally with hard partition. For a given sample, the convolution operations at different nodes of the same layer can not be activated at the same time. Therefore, the number of modalities equals to the number of the leaf nodes, which has to be determined in advance. However, the combination of modality-specific variations is unpredictable and much more complex in reality, and the tree-based CNN is not robust enough to resolve such a challenging issue.

- Adaptive Learning introduced in Chapter 5 requires the re-training of the classifier each time a confident track is selected. This scheme of update is applicable for face recognition problem of a small scale. However, the objective of the proposed method is to mimic the learning behavior of human, i.e., long-term never-ending learning. With the current approach, the computation and storage requirement will increase exponentially as more and more samples or tracks are selected iteration by iteration. Therefore, a more efficient learning scheme will be interesting.

Accordingly, further exploration of research directions on the following perspectives will be interesting and promising.

- **Joint Learning of DMM and CFN**. DMM is designed to find the patch correspondences, while CFN aims at learning the part-based facial features. Chapter 3 optimizes the two networks separately. A more proper approach could be a unified network which locates the key patches at the front and extracts the patch-specific representation in the following layers. By optimizing the unified network, both the patch selection and feature learning are learnt with regard to the given problem directly. Besides, it would also be interesting to substitute the standard CNN of CFN with c-CNN proposed in Chapter 4.

- **Conditional CNN with Dynamic Partition**. Theoretically, the routing of sample through c-CNN could be more flexible and dynamic than tree-based c-CNN.

In particular, the number of activated kernels in each layer should be dynamic for each sample, and the activations of convolutional kernels do not have to be achieved by group. Dynamic partition of activated kernels will render more complex and diverse routes. As each route represents one channel of feature extraction corresponding to one modality, the number of modalities that c-CNN can include is largely improved, so is the generalization performance of the network.

- **Online Adaptive Learning**. Online learning, aims at sequentially updating the classifier, becomes essential when the dataset is too large to be processed in a single run. The online classifier are expected to update incrementally without exponentially growing memory and computational cost such that a life-long learning system is possible. The "sequential" or "incremental" property makes online learning a good option for applications in real world scenario. The integration of online learning with adaptive learning will bring the proposed method more close to human behaviors.

- **Deep Semi-Supervised Learning**. Most of deep neural networks are learnt in a supervised manner. The unlabeled data are usually utilized in the pre-training of auto-encoder and deep belief network, which aims at providing a good initialization for the later supervised fine-tuning step. Moreover, the unlabeled data used in pre-training are usually not directly related to the given problem. The idea of semi-supervised deep learning has not been well explored yet. C-CNN does have a supervised cost and a unsupervised cost at each tree node, but the unsupervised cost is defined in batches, and thus is only a simple approximation of the global semi-supervised constraint. Many research works have proven that utilizing the unlabeled in the learning of classifier usually brings considerable improvements. Therefore, the combination of DNN and SSL will be a promising direction. Moreover, deep semi-supervised learning can be applied more easily in the adaptive learning framework so as to improve the performance further.

References

[Ahonen et al., 2006] Ahonen, T., Hadid, A., and Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041. 38

[Arandjelovic and Zisserman, 2005] Arandjelovic, O. and Zisserman, A. (2005). Automatic face recognition for film character retrieval in feature-length films. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 860–867. 92

[Ballan et al., 2010] Ballan, L., Bertini, M., Del Bimbo, A., and Serra, G. (2010). Semantic annotation of soccer videos by visual instance clustering and spatial/temporal reasoning in ontologies. *Multimedia Tools and Applications*, 48(2):313–337. 92

[Bauml et al., 2013] Bauml, M., Tapaswi, M., and Stiefelhagen, R. (2013). Semi-supervised Learning with Constraints for Person Identification in Multimedia Data. *IEEE Conference on Computer Vision and Pattern Recognition*. 96

[Belhumeur et al., 1997] Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. (1997). Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720. 4, 16

[Belkin et al., 2006] Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434. 9, 93, 95, 103, 104, 106

[Berg et al., 2004] Berg, A., Edwards, J., Maire, M., White, R., Teh, Y., Learned-Miller, E., and Forsyth, D. (2004). Names and faces in the news. *IEEE Conference on Computer Vision and Pattern Recognition*. 92, 95

[Bertini et al., 2006] Bertini, M., Del Bimbo, A., and Nunziati, W. (2006). Automatic detection of player's identity in soccer videos using faces and text cues. In *ACM international conference on Multimedia*, MULTIMEDIA '06, pages 663–666. 92

[Blum and Mitchell, 1998] Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 92–100, New York, NY, USA. 96, 97

[Bosch et al., 2007] Bosch, A., Zisserman, A., and Munoz, X. (2007). Image classification using random forests and ferns. In *International Conference on Computer Vision*, pages 1–8. IEEE. 20

[Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *PROCEEDINGS OF THE 5TH ANNUAL ACM WORKSHOP ON COMPUTATIONAL LEARNING THEORY*. 20

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*. 20, 79

[Bulo and Kontschieder, 2014] Bulo, S. R. and Kontschieder, P. (2014). Neural decision forests for semantic image labelling. In *IEEE Conference on Computer Vision and Pattern Recognition*. 70, 71, 77

[Chen and Grauman, 2013] Chen, C.-Y. and Grauman, K. (2013). Watching unlabeled video helps learn new human actions from very few labeled snapshots. *IEEE Conference on Computer Vision and Pattern Recognition*. 96

[Chen et al., 2012] Chen, D., Cao, X., Wang, L., Wen, F., and Sun, J. (2012). Bayesian face revisited: A joint formulation. In *European Conference on Computer Vision*, pages 566–579. Springer. 4

[Chen et al., 2013] Chen, D., Cao, X., Wen, F., and Sun, J. (2013). Blessing of dimensionality: High-dimensional feature and its efiňĄcient compression for face veriiňĄcation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 17, 85, 88

[Chen et al., 2014] Chen, D., Ren, S., Wei, Y., Cao, X., and Sun, J. (2014). Joint cascade face detection and alignment. In *European Conference on Computer Vision*, pages 109–122. Springer. 2

[Cherniavsky et al., 2010] Cherniavsky, N., Laptev, I., Sivic, J., and Zisserman, A. (2010). Semi-supervised learning of facial attributes in video. *European Conference on Computer Vision Workshops*. 9, 96

[Choi et al., 2013] Choi, J., Ali, M. R., Larry, F., and Davis, S. (2013). Adding unlabeled samples to categories by learned attributes. *IEEE Conference on Computer Vision and Pattern Recognition*. 96

[Chopra et al., 2005] Chopra, S., Hadsell, R., and Lecun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*. 40, 42

[Collobert et al., 2006] Collobert, R., Sinz, F., Weston, J., and Bottou, L. (2006). Large scale transductive SVMs. *Journal of Machine Learning Research*, 7:1687–1712. 29, 120

[Cristianini and Shawe-Taylor, 2000] Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press. 108

[Cui et al., 2013] Cui, Z., Li, W., Xu, D., Shan, S., and Chen, X. (2013). Fusing robust face region descriptors via multiple metric learning for face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*. 21, 34, 56

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 66

[Dasgupta and Freund, 2008] Dasgupta, S. and Freund, Y. (2008). Random projection trees and low dimensional manifolds. In *ACM Symposium on Theory of computing*. 74

[Daugman, 1985] Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A-Optics Image Science and Vision*. 6, 114

[Davis et al., 2007] Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. (2007). Information-theoretic metric learning. In *International Conference on Machine Learning*, pages 209–216. ACM. 21

[Domingos and Pazzani, 1997] Domingos, P. and Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130. 19

[Erhan et al., 2010] Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11:625–660. 51

[Everingham et al., 2006] Everingham, M., Sivic, J., and Zisserman, A. (2006). 'Hello! My name is... buffy' - automatic naming of characters in tv video. In *British Machine Vision Conference*, pages 889–908. 95, 114

[Fanello et al., 2014] Fanello, S. R., Keskin, C., Kohli, P., Izadi, S., Shotton, J., Criminisi, A., Pattacini, U., and Paek, T. (2014). Filter forests for learning data-dependent convolutional kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*. 71, 77

[Farabet et al., 2013] Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 7, 21, 34, 39

[Farfade et al., 2015] Farfade, S. S., Saberian, M., and Li, L.-J. (2015). Multi-view face detection using deep convolutional neural networks. *arXiv preprint arXiv:1502.02766*. 3

[Fathi et al., 2011] Fathi, A., Balcan, M. F., Ren, X., and Rehg, J. M. (2011). Combining self training and active learning for video segmentation. *British Machine Vision Conference*. 96

[Fette et al., 2007] Fette, I., Sadeh, N., and Tomasic, A. (2007). Learning to detect phishing emails. In *International conference on World Wide Web*, pages 649–656. ACM. 20

[Freund and Schapire, 1999] Freund, Y. and Schapire, R. (1999). A short introduction to boosting. In *International Joint Conferences on Artificial Intelligence*. 79

[Gall and Lempitsky, 2013] Gall, J. and Lempitsky, V. (2013). Class-specific hough forests for object detection. In *Decision forests for computer vision and medical image analysis*, pages 143–157. Springer. 20

[Gao and Ma, 2011] Gao, G. and Ma, H. (2011). Accelerating shot boundary detection by reducing spatial and temporal redundant information. In *IEEE International Conference on Multimedia and Expo*, pages 1–6. 112

[Geladi and Kowalski, 1986] Geladi, P. and Kowalski, B. R. (1986). Partial least-squares regression: a tutorial. *Analytica Chimica Acta*. 66

[Ghiasi and Fowlkes, 2014] Ghiasi, G. and Fowlkes, C. C. (2014). Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1899–1906. IEEE. 3

[Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 66, 70

[Goldman and Zhou, 2000] Goldman, S. and Zhou, Y. (2000). Enhancing supervised learning with unlabeled data. *International Conference on Machine Learning*. 97

[Gonzalez and Woods, 2002] Gonzalez, R. C. and Woods, R. E. (2002). Digital image processing. 42

[Grabner and Bischof, 2006] Grabner, H. and Bischof, H. (2006). Online boosting and vision. *IEEE Conference on Computer Vision and Pattern Recognition*. 97

[Grabner et al., 2008] Grabner, H., Leistner, C., and Bischof, H. (2008). Semi-supervised on-line boosting for robust tracking. In *European Conference on Computer Vision*, ECCV '08, pages 234–247. 97

[Gross et al., 2010] Gross, R., Matthews, I., Cohn, J., Kanade, T., and Baker, S. (2010). Multi-pie. *Image and Vision Computing*. 80

[Grossberg, 2013] Grossberg, S. (2013). Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world. *Neural Networks*, 37:1–47. 94, 97, 99

[Guillaumin et al., 2009] Guillaumin, M., Verbeek, J., and Schmid, C. (2009). Is that you? Metric learning approaches for face identification. In *International Conference on Computer Vision*, pages 498–505. IEEE. 21

[Hardoon et al., 2004] Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*. 66

[Hinton and Salakhutdinov, 2006] Hinton, G. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*. 21, 24, 39

[Hinton et al., 2012] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*. 54

[Hu et al., 2013] Hu, J., Lu, J., and Tan, Y.-P. (2013). Discriminative deep metric learning for face verification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*. 8, 18, 41, 56

[Hu et al., 2015] Hu, J., Lu, J., Yuan, J., and Tan, Y.-P. (2015). Large margin multi-metric learning for face and kinship verification in the wild. In *Asian Conference on Computer Vision*, pages 252–267. Springer. 57

[Hua and Akbarzadeh, 2009] Hua, G. and Akbarzadeh, A. (2009). A robust elastic and partial matching metric for face recognition. In *International Conference on Computer Vision*. 42

[Huang et al., 2007a] Huang, C., Ai, H., Li, Y., and Lao, S. (2007a). High-performance rotation invariant multiview face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):671–686. 3

[Huang et al., 2012a] Huang, G., Mattar, M., Lee, H., and Learned-Miller, E. (2012a). Learning to align from scratch. In *Advances in Neural Information Processing Systems*. 58

[Huang et al., 2007b] Huang, G., Ramesh, M., Berg, T., and Learned-Miller, E. (2007b). Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments. Technical report. 8, 33, 80, 85

[Huang et al., 2012b] Huang, G. B., Lee, H., and Learned-Miller, E. (2012b). Learning hierarchical representations for face verification with convolutional deep belief networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 7, 21, 34, 39, 40

[Hussain et al., 2012] Hussain, S., Napoleon, T., and Jurie, F. (2012). Face recognition using local quantized patterns. In *British Machine Vision Conference*. 38

[Jordan and Jacobs, 1994] Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*. 70

[Kan et al., 2014] Kan, M., Shan, S., Chang, H., and Chen, X. (2014). Stacked progressive auto-encoders (spae) for face recognition across poses. In *IEEE Conference on Computer Vision and Pattern Recognition*. 66, 70

[Kan et al., 2012] Kan, M., Shan, S., Zhang, H., Lao, S., and Chen, X. (2012). Multi-view discriminant analysis. In *European Conference on Computer Vision*. 69

[Kim et al., 2008] Kim, M., Kumar, S., Pavlovic, V., and Rowley, H. (2008). Face tracking and recognition with visual constraints in real-world videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE. 122

[Kim et al., 2003] Kim, T.-K., Kim, H., Hwang, W., Kee, S., and Kittler, J. (2003). Independent component analysis in a facial local residue space. In *IEEE Conference on Computer Vision and Pattern Recognition*. 38

[Kim et al., 2005] Kim, T.-K., Kim, H., Hwang, W., and Kittler, J. (2005). Component-based LDA face description for image retrieval and MPEG-7 standardisation. *Image and Vision Computing*, 23(7):631–642. 38

[Kim and Kittler, 2005] Kim, T.-K. and Kittler, J. (2005). Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 4, 68

[Kontschieder et al., 2015] Kontschieder, P., Fiterau, M., Criminisi, A., and Rota Bulo, S. (2015). Deep neural decision forests. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1467–1475. 71

[Kouzani et al., 2007] Kouzani, A., Nahavandi, S., and Khoshmanesh, K. (2007). Face classification by a random forest. In *IEEE Region 10 Conference: TENCON 2007*, pages 1–4. IEEE Xplore. 4

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 7, 21, 34, 39, 40, 43, 69

[Kuettel et al., 2012] Kuettel, D., Guillaumin, M., and Ferrari, V. (2012). Segmentation propagation in imageNet. *European Conference on Computer Vision*. 96

[Kumar et al., 2009] Kumar, N., Berg, A. C., Belhumeur, P. N., and Nayar, S. K. (2009). Attribute and simile classifier for face verification. In *International Conference on Computer Vision*. 8

[Li et al., 2013] Li, H., Hua, G., Lin, Z., Brandt, J., and Yang, J. (2013). Probabilistic elastic matching for pose variant face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4, 17, 34, 35, 36, 39, 41, 45, 46, 56, 59, 60, 85, 88

[Li et al., 2015] Li, H., Hua, G., Shen, X., Lin, Z., and Brandt, J. (2015). Eigen-pep for video face recognition. In *Asian Conference on Computer Vision*, pages 17–33. Springer. 57

[Li et al., 2012] Li, S., Liu, X., Chai, X., Zhang, H., Lao, S., and Shan, S. (2012). Morphable displacement field based image matching for face recognition across pose. In *European Conference on Computer Vision*. 69

[Li et al., 2002] Li, S. Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., and Shum, H. (2002). Statistical learning of multi-view face detection. In *European Conference on Computer Vision*, pages 67–81. Springer. 3

[Li and Guo, 2013] Li, X. and Guo, Y. (2013). Adaptive active learning for image classification. *IEEE Conference on Computer Vision and Pattern Recognition*. 96

[Liao et al., 2013] Liao, Q., Leibo, J. Z., Mroueh, Y., and Poggio, T. (2013). Can a biologically-plausible hierarchy effectively replace face detection, alignment, and recognition pipelines? *arXiv preprint arXiv:1311.4082*. 40

[Liao et al., 2007] Liao, S., Zhu, X., Lei, Z., Zhang, L., and Li, S. Z. (2007). Learning multi-scale block local binary patterns for face recognition. In *Advances in Biometrics*, pages 828–837. Springer. 17

[Lin et al., 2013] Lin, M., Chen, Q., and Yan, S. (2013). Network in network. In *ArXiv e-prints*. 66, 69, 70

[Liu et al., 2007] Liu, C., Shum, H.-Y., and Freeman, W. T. (2007). Face hallucination: Theory and practice. *International Journal of Computer Vision*. 66, 69

[Liu and Wechsler, 2002] Liu, C. and Wechsler, H. (2002). Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing*, 11(4):467–476. 34, 38

[Liu and Wang, 2007] Liu, Z. and Wang, Y. (2007). Major cast detection in video using both speaker and face information. *IEEE Transactions on Multimedia*, 9(1):89–101. 95

[Lowe and G, 1999] Lowe and G, D. (1999). Object recognition from local scale-invariant features. *International Conference on Computer Vision*. 4, 6, 17, 66, 114

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110. 34

[Lu et al., 2015] Lu, J., Wang, G., Deng, W., and Jia, K. (2015). Reconstruction-based metric learning for unconstrained face verification. *IEEE Transactions on Information Forensics and Security*, 10(1):79–89. 57

[Luo et al., 2012] Luo, P., Wang, X., and Tang, X. (2012). Hierarchical face parsing via deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 38

[McClosky et al., 2006] McClosky, D., Charniak, E., and Johnson, M. (2006). Effective self-training for parsing. *The 2006 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 94

[Melacci and Belkin, 2011] Melacci, S. and Belkin, M. (2011). Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research*, 12:1149–1184. 29, 115

[Mendez-Vazquez et al., 2013] Mendez-Vazquez, H., Martinez-Diaz, Y., and Chai, Z. (2013). Volume structured ordinal features with background similarity measure for video face recognition. In *International Conference on Biometrics*. IEEE. 56

[Mignon and Jurie, 2012] Mignon, A. and Jurie, F. (2012). CMML: A new metric learning approach for cross modal matching. In *Asian Conference on Computer Vision*. 4

[Minh et al., 2013] Minh, H. Q., Bazzani, L., and Murino, V. (2013). A unifying framework for vector-valued manifold regularization and multi-view learning. In *International Conference on Machine Learning*, volume 28, pages 100–108. 97

[Mitchell, 1999] Mitchell, T. (1999). The role of unlabeled data in supervised learning. In *Proceedings of the sixth International Colloquium on Cognitive Science*, pages 2–11. Citeseer. 27, 28

[Nair and Hinton, 2010] Nair, V. and Hinton, G. (2010). Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*. 7, 21, 23, 34, 39, 42, 59, 60

[Ojala et al., 1996] Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59. 4, 34, 38

[Ojala et al., 2002a] Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002a). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 66

[Ojala et al., 2002b] Ojala, T., Pietikinen, M., and Menp, T. (2002b). Multiresolution grayscale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 6, 17, 114

[Ouyang and Wang, 2013] Ouyang, W. and Wang, X. (2013). Joint deep learning for pedestrian detection. In *International Conference on Computer Vision*. IEEE. 40

[Person, 1901] Person, K. (1901). On lines and planes of closest fit to system of points in space. 16

[Pinto et al., 2009] Pinto, N., DiCarlo, J., and Cox, D. (2009). How far can you get with a modern face recognition test set using only simple features? In *IEEE Conference on Computer Vision and Pattern Recognition*. 60

[Prest et al., 2012] Prest, A., Leistner, C., Civera, J., Schmid, C., and Ferrari, V. (2012). Learning object class detectors from weakly annotated video. *IEEE Conference on Computer Vision and Pattern Recognition*. 96

[Saffari et al., 2010] Saffari, A., Leistner, C., Godec, M., and Bischof, H. (2010). Robust multi-view boosting with priors. In *European Conference on Computer Vision*, volume 6313 of *Lecture Notes in Computer Science*, pages 776–789. 97

[Sagonas et al., 2013] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2013). A Semi-automatic methodology for facial landmark annotation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 896–903. 3

[Satoh et al., 1999] Satoh, S., Nakamura, Y., and Kanade, T. (1999). Name-it: naming and detecting faces in news videos. *IEEE Multimedia*. 92, 95

[Scholkopf et al., 2001] Scholkopf, B., Herbrich, R., and Smola, A. (2001). A generalized representer theorem. In *Computational Learning Theory*, volume 2111 of *Lecture Notes in Computer Science*, pages 416–426. 104

[Sharma et al., 2012] Sharma, A., Al Haj, M., Choi, J., Davis, L. S., and Jacobs, D. W. (2012). Robust pose invariant face recognition using coupled latent space discriminant analysis. *Computer Vision and Image Understanding*. 69

[Sharma and Jacobs, 2011] Sharma, A. and Jacobs, D. W. (2011). Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4, 69

[Shrivastava et al., 2012] Shrivastava, A., Singh, S., and Gupta, A. (2012). Constrained semi-supervised learning using attributes and comparative attributes. *European Conference on Computer Vision*. 94, 96, 102

[Sikka et al., 2012] Sikka, K., Wu, T., Susskind, J., and Bartlett, M. (2012). Exploring bag of words architectures in the facial expression domain. In *European Conference on Computer Vision Workshops*. 38

[Simonyan et al., 2013] Simonyan, K., Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2013). Fisher vector faces in the wild. In *British Machine Vision Conference*. 17, 18, 34, 35, 38, 39, 41, 45, 59, 60, 82, 83, 85, 88

[Sun, 2011] Sun, S. (2011). Multi-view laplacian support vector machines. In *Advanced Data Mining and Applications*, Lecture Notes in Computer Science, pages 209–222. 108

[Sun et al., 2014a] Sun, Y., Chen, Y., Wang, X., and Tang, X. (2014a). Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996. 35, 40

[Sun et al., 2013a] Sun, Y., Wang, X., and Tang, X. (2013a). Deep convolutional network cascade for facial point detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 7, 21, 34, 39, 40

[Sun et al., 2013b] Sun, Y., Wang, X., and Tang, X. (2013b). Hybrid deep learning for face verification. In *International Conference on Computer Vision*. 2

[Sun et al., 2014b] Sun, Y., Wang, X., and Tang, X. (2014b). Deep learning face representation from predicting 10,000 classes. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2, 8, 35, 40, 66, 70

[Taigman et al., 2014] Taigman, Y., Yang, M., Marc'Aurelio, R., and Wolf, L. (2014). DeepFace: closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2, 8, 35, 40

[Tan and Triggs, 2010] Tan, X. and Triggs, B. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650. 38, 42, 53

[Tapaswi et al., 2012] Tapaswi, M., Bäuml, M., and Stiefelhagen, R. (2012). "Knock! Knock! Who is it?" Probabilistic person identification in TV series. In *IEEE Conference on Computer Vision and Pattern Recognition*. 95

[Tu, 2005] Tu, Z. (2005). Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *International Conference on Computer Vision*. 70

[Turk and Pentland, 1991] Turk, M. A. and Pentland, A. P. (1991). Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4, 16, 38

[Tzelepis et al., 2015] Tzelepis, C., Galanopoulos, D., Mezaris, V., and Patras, I. (2015). Learning to detect video events from zero or very few video examples. *Image and Vision Computing*. 105

[Tzelepis et al., 2013] Tzelepis, C., Gkalelis, N., Mezaris, V., and Kompatsiaris, I. (2013). Improving event detection using related videos and relevance degree support vector machines. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 673–676. ACM. 105

[Vapnik and Vapnik, 1998] Vapnik, V. N. and Vapnik, V. (1998). *Statistical learning theory*, volume 1. Wiley New York. 20

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–518. 3

[Wang et al., 2004] Wang, H., Li, S., and Wang, Y. (2004). Face recognition under varying lighting conditions using self quotient image. In *IEEE International Conference on Automatic Face and Gesture Recognition*. 42

[Wang et al., 2011] Wang, S., Lu, H., F., Y., and M.H., Y. (2011). Superpixel tracking. In *International Conference on Computer Vision*, pages 1323–1330. 112

[Wang and Tang, 2009] Wang, X. and Tang, X. (2009). Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 66, 69

[Weston et al., 2000] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2000). Feature selection for SVMs. In *Advances in Neural Information Processing Systems*. 49, 50

[Wolf et al., 2011] Wolf, L., Hassner, T., and Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 8, 33, 56

[Wolf et al., 2008] Wolf, L., Hassner, T., and Taigman, Y. (2008). Descriptor based methods in the wild. In *European Conference on Computer Vision Workshops*. 60

[Wolf and Levy, 2013] Wolf, L. and Levy, N. (2013). The SVM-minus similarity score for video face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 56

[Wright and Hua, 2009] Wright, J. and Hua, G. (2009). Implicit elastic matching with random projections for pose-variant face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 36

[Wu and Nevatia, 2007] Wu, B. and Nevatia, R. (2007). Cluster boosted tree classifier for multi-view, multi-pose object detection. In *International Conference on Computer Vision*. 70

[Xiong et al., 2015a] Xiong, C., Liu, L., Zhao, X., Yan, S., and Kim, T.-K. (2015a). Convolutional fusion network for face verification in the wild. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1. 57

[Xiong et al., 2015b] Xiong, C., Zhao, X., Tang, D., Jayashree, K., Shuicheng, Y., and Kim, T.-K. (2015b). Conditional convolutional neural network for modality-aware face recognition. In *IEEE International Conference on Computer Vision*. 71

[Yan et al., 2014] Yan, J., Lei, Z., Wen, L., and Li, S. Z. (2014). The fastest deformable part model for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2497–2504. IEEE. 3

[Yan et al., 2013] Yan, J., Lei, Z., Yi, D., and Li, S. Z. (2013). Learn to combine multiple hypotheses for accurate face alignment. In *International Conference on Computer Vision Workshops*, pages 392–396. 3

[Yan et al., 2006] Yan, R., Zhang, J., Yang, J., and Hauptmann, A. G. (2006). A discriminative learning framework with pairwise constraints for video object classification. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28. 96

[Yan et al., 2008] Yan, S., Shan, S., Chen, X., and Gao, W. (2008). Locally Assembled Binary (LAB) feature with feature-centric cascade for fast and accurate face detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. 17

[Yang et al., 2007] Yang, A. Y., Wright, J., Ma, Y., and Sastry, S. S. (2007). Feature selection in face recognition: A sparse representation perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 16

[Yang et al., 2015] Yang, H., Mou, W., Zhang, Y., Patras, I., Gunes, H., and Robinson, P. (2015). Face alignment assisted by head pose estimation. *arXiv preprint arXiv:1507.03148*. 3

[Yang et al., 2011] Yang, M., Zhang, L., Yang, J., and Zhang, D. (2011). Robust sparse coding for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 625–632. IEEE. 16

[Yang et al., 2013] Yang, Y., Shu, G., and Shah, M. (2013). Semi-supervised learning of feature hierarchies for object detection in a video. *IEEE Conference on Computer Vision and Pattern Recognition*. 96

[Yarowsky, 1995] Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196. 27, 94

[Zhai et al., 2012] Zhai, Y., Tan, M., Tsang, I. W., and Ong, Y. (2012). Discovering support and affiliated features from very high dimensions. In *International Conference on Machine Learning*. 49, 50

[Zhang et al., 2005] Zhang, W., Shan, S., Gao, W., Chen, X., and Zhang, H. (2005). Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. In *International Conference on Computer Vision*, volume 1, pages 786–791. IEEE. 17

[Zhang et al., 2006] Zhang, X., Gao, Y., and Leung, M. K. H. (2006). Automatic texture synthesis for face recognition from single views. In *International Conference on Pattern Recognition*. 69

[Zhang et al., 2012] Zhang, X., Zhang, L., Wang, X.-J., and Shum, H.-Y. (2012). Finding celebrities in billions of web images. *IEEE Transactions on Multimedia*, 14(4):995–1007. 92

[Zhao and Pietikainen, 2007] Zhao, G. and Pietikainen, M. (2007). Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928. 17

[Zhao et al., 2008] Zhao, M., Yagnik, J., Adam, H., and Bau, D. (2008). Large scale learning and recognition of faces in web videos. *IEEE International Conference on AutomaticFace and Gesture Recognition*. 92

[Zhao et al., 2013] Zhao, X., Kim, T.-K., and Luo, W. (2013). Unified face analysis by iterative multi-output random forests. In *IEEE Conference on Computer Vision and Pattern Recognition*. 38, 70

[Zhou et al., 2003] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Scholkopf, B. (2003). Learning with local and global consistency. *Advances in Neural Information Processing Systems*. 93, 95

[Zhou et al., 2013] Zhou, E., Fan, H., Cao, Z., Jiang, Y., and Yin, Q. (2013). Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *International Conference on Computer Vision Workshops*, pages 386–391. IEEE. 3

[Zhu et al., 2012] Zhu, P., Zhang, L., Hu, Q., and Shiu, S. (2012). Multi-scale patch based collaborative representation for face recognition with margin distribution optimization. In *European Conference on Computer Vision*. 38

[Zhu, 2005] Zhu, X. (2005). Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison. 27

[Zhu et al., 2003a] Zhu, X., Ghahramani, Z., Lafferty, J., and Others (2003a). Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning*, volume 3, pages 912–919. 9, 27, 29

[Zhu et al., 2003b] Zhu, X., Lafferty, J., and Ghahramani, Z. (2003b). Semi-supervised learning: from Gaussian fields to Gaussian processes. *International Conference on Machine Learning*. 93, 95

[Zhu et al., 2015] Zhu, X., Lei, Z., Yan, J., Yi, D., and Li, S. Z. (2015). High-fidelity pose and expression normalization for face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 787–796. 17

[Zhu and Ramanan, 2012] Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2879–2886. IEEE. 2

[Zhu et al., 2013] Zhu, Z., Luo, P., Wang, X., and Tang, X. (2013). Deep learning identity-preserving face space. In *International Conference on Computer Vision*. 66, 70, 81, 82, 83

[Zhu et al., 2014] Zhu, Z., Luo, P., Wang, X., and Tang, X. (2014). Multi-view perceptron: a deep model for learning face identity and view representations. In *Advances in Neural Information Processing Systems*. 66, 70